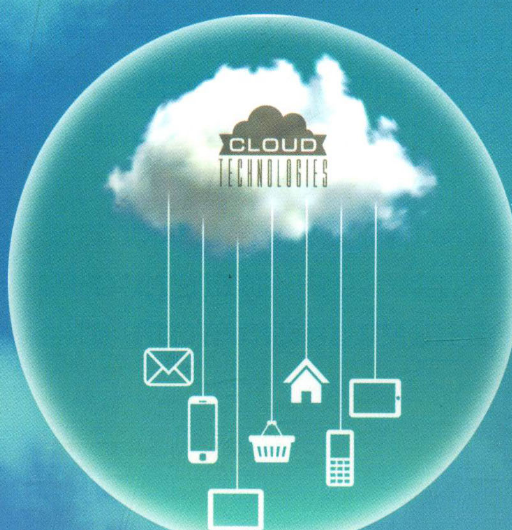
CRC Press
Taylor & Francis Group

云计算 开发与安全

Developing and Securing the Cloud

[美] 巴瓦尼 · 杜莱辛甘（Bhavani Thuraisingham）著
林果园 等译



开发安全云的实用指南

详述基本概念、思想、方法和实例

机械工业出版社
CHINA MACHINE PRESS

VISIT…

LANZAROTE Caliente.com

作者简介

Bhavani Thuraisingham博士是得克萨斯大学达拉斯分校(UTD)埃里克乔森工程与计算机科学学院Louis A. Beecherl,Jr.杰出教授，以及UTD网络安全研究和教育研究所执行董事。她目前从事的是网络安全、云计算和大数据分析的集成性研究。

Thuraisingham博士是美国电气电子工程师学会（IEEE）、美国科学促进会（AAAS）和英国计算机学会的会士（Fellow）。她发表过100余篇期刊论文、200余篇会议论文，出版过12本图书，完成超过100场主旨演讲。她已经拥有8项专利(3项待定)，并且是知识与安全分析有限责任公司的创始人。

云计算开发与安全

Developing and Securing the Cloud

[美] 巴瓦尼·杜莱辛甘（Bhavani Thuraisingham） 著

林果园 等译

机 械 工 业 出 版 社

本书主要论述云计算开发与安全的基本概念、思想、方法和实例。全书主要由9个部分组成，内容涉及相关支撑技术、云计算概念和开发技术、云的应用、云计算安全、云计算与安全的教育和研究等相关技术及其实例、实验系统等。

本书是一本不可多得的将云计算开发与安全紧密结合的图书。通过风趣且智慧的语言讲解抽象的概念，用实例和插图帮助读者理解其中的含义，同时又不失论述的严谨性，使读者易于理解和掌握。

本书可作为高等院校信息技术相关专业的本科生和研究生的教材，也适合从事云计算设计开发、安全维护和应用等专业技术人员，使用云的企业管理人员，云计算与安全爱好者阅读。

译　者　序

云计算是一种新计算模型，它将计算任务分布在大量计算机构成的资源池上，使用户能够按需获取计算能力、存储空间和信息服务等。有人将这种模式比喻为从单台发电机供电模式转向了电厂集中供电的模式。它意味着计算能力也可以作为一种商品进行流通，就像煤气、水和电一样，取用方便，费用低廉，但与传统商品最大的不同在于，它是通过互联网进行传输的。

云计算由多种传统技术相互融合衍生而来，涉及了网格、分布式、并行及效用等多种计算方法，还涉及了网络存储、虚拟化及负载均衡等多种技术，这种计算服务方式，以其宽带互联、资源池共享、弹性配置、按需服务和按量收费等独特优势，在各行业应用中快速兴起。

云计算与以往的计算模式安全风险不同。云计算环境下，信息安全问题更严重、更突出，核心的问题有两个方面。首先是以前的信息系统都是自己建的，或者托管，在安全资源和基础设备方面有可控性。在云计算的环境下，是由服务提供商统管 IT 资源和基础设施，用户自己无法管理和控制。第二个问题就是更大规模异构共享和虚拟动态的运行环境难以控制，云计算是属于动态变化的计算环境，这个运行环境在某种意义上是无序的。

因此，我们在使用云、开发云的同时，也需要关注它的安全问题。市面上关于云计算开发的书籍琳琅满目，但是它们大多是阐述开发理论和技术，没有涉及安全。关于云计算安全的书籍又过多地偏重安全，没有考虑到云计算本身开发利用的实际情况。本书是不可多得的好书，它很好地解决了这个问题，是一本将云开发与安全紧密结合、理论与实践相互融合的书籍。

作为云计算开发与安全的优秀之作，本书有以下几个特点：

1. 内容全面：全书共分 9 个部分，内容涉及相关支撑技术（数据、信息、知识管理、分布式计算、访问控制等）、云计算概念和开发技术（服务模型、语义 Web、云存储管理、数据管理、虚拟机监控和管理等）、云的应用（社交网络、电子商务、供应链、工作流、恶意软件与内部威胁检测等）、云计算安全技术（服务安全、语义 Web 安全、查询处理安全及信任、隐私、信息共享安全等）、云计算与安全的教育和研究等相关技术及其实验系统。可以说覆盖了云计算开发与安全的各个重要方面。

2. 易于理解：书中所提及的原理和概念都有详细解释，并配有很多实例和插图帮助读者理解，以充实的内容在抽象概念和实际实现之间架设了桥梁。本书讨论了云计算开发与安全中的基本概念与算法，提供了相关的具体实例，如移动云、多媒

体云、基于云的社交网络、云计算实验系统、面向安全应用的云实验系统、云计算安全实验系统等。为读者深入浅出地学习和理解云计算与安全提供了坚实的理论基础，用风趣而智慧的语言讲解许多抽象的概念。

3. 内容新颖：本书不仅加入了快速发展的云计算开发和安全的新概念，还将作者团队在云计算开发与安全方面实现的几项开创性工作纳入到书中。其中：第一个根据 OMT（Object Modeling Technique，对象建模技术）模型检查 OOAD（Object Oriented Andysis and Design，面向对象的分析和设计）的安全性。开发了一个安全的对象模型、安全的动态模型和安全的功能模型。第一个提出使用 HDFS 中的平面文件来提供 RDF 数据的存储方案，以及用于回答 SPARQL 查询的 MapReduce 作业确定性算法。设计和实施的 CAISS 是第一个支持以云为中心的 AIS 的系统。

综合来看，本书具有内容新颖、全面、实用性指导性强等特点，不仅是从事云计算开发与安全等专业人士的必备之书，同时也是高等院校相关专业师生教学的最佳教材，由衷地希望所有读者都能从本书中充分体会到云计算开发与安全的精髓，并在今后的相关工作中游刃有余。

本书的翻译工作由林果园总体组织负责，审校全书译稿，佟晶晶参加了本书的文字润色工作。贺寰烨、王雨琪、朱琪、郭练等参加了第 1 ~ 4 章的翻译工作；顾浩、靖元、孟科技、田凯等参加了第 5 ~ 7 章的翻译工作；徐武兴、杨飞、路训禄、唐乾等参加了第 8 ~ 12 章的翻译工作；何望、朱洁、周娜等参加了第 13 ~ 15 章的翻译工作；王丹茹、薛良勇、臧平平、王子恩、郝丹丹、孔珍珍等参加了第 16 ~ 21 章的翻译工作；许尔旭、马栋、戚中奎等参加了第 22 ~ 24 章的翻译工作；韩浩、贺珊、唐久涛、王国辉等参加了第 25 ~ 28 章的翻译工作；李帅、李超、王轩等参加了第 29 ~ 31 章的翻译工作；黄琪、别玉玉、刘凯、齐乐等参加了第 32 ~ 35 章的翻译工作；刘博文、黄芸、仇永泉、李国强等参加了附录的翻译工作。

本书的翻译在力求忠实于原著的同时，又力求符合汉语习惯。我们在许多云计算开发与安全的专业术语后面，注明了英文原文，这一方面是为了方便读者能对照理解，为今后的学习打下基础；另一方面，也是为了避免因中文译法带来的歧义，从而节省读者宝贵的阅读时间。

在本书的翻译过程中得到了世界很多同行的帮助，在此表示深深的谢意，同时也非常感谢机械工业出版社的编辑们给予我们的支持，由于种种原因书中难免存在错误和不妥之处，恳请读者批评指正。

译者

原 书 前 言

背景

随着信息系统技术的不断发展，许多业务领域的各种应用日益趋于计算机化。在许多组织机构中，数据已经成为其关键资源。有效地访问数据、共享数据以及从数据中提取信息并利用信息已经成为迫切的需要。因此，我们不仅需要努力整合分散在多个站点上的各种数据源，而且从这些数据库中提取信息模式和趋势也显得十分重要。这些数据源可以是由数据库管理系统所管理的数据库，也可以是来自于多种数据源的在数据仓库中存储的数据。

20 世纪 90 年代中期，随着万维网（World Wide Web，WWW）的发展，对数据、信息和知识的有效管理产生了更大需求。在这个阶段，消费者书籍销售商的概念已通过网络进行了数字化体现。按照这种方式，许多服务有了网络的支持，消费者可以通过服务提供商的网站请求服务，而服务提供商提供被请求的服务。这种服务可以是从航空公司预订机票或者是从书籍销售商那里购买一本书，这种由网络支持的服务被称为网络服务。需要注意的是，这种服务不一定必须通过网络提供。消费者可以向服务提供商发送电子邮件请求服务。这种服务就是计算机支持的服务，由计算机支持的服务大部分工作都集中在网络服务上。

这种服务模式已经演变成提供计算服务的基础设施、软件、数据库以及各种应用。例如，像从电力公司获得供电服务一样，我们也可以从服务提供商那里获得计算服务，这种能力就产生了云计算的概念。正如美国国家标准与技术研究院对云计算的定义：云计算是实现无处不在、方便、按需网络访问共享的可配置计算资源池（如网络、服务器、存储、应用程序和服务）的模型，可以通过最少的管理努力或服务提供商的交互来快速地提供和发布。

在过去 5 年中，云计算的发展迅速，目前已经有好几家公司提供基础设施软件、应用程序和计算平台作为服务。

随着对数据和信息管理需求的不断增加，维护数据库、应用程序和信息系统的安全性至关重要，因此必须保护数据和信息免受未经授权的访问以及恶意破坏。随着云计算的出现，由于云数据通常是由第三方管理，所以保护数据和信息尤为重要，因此需要有效的机制来保护云计算安全。

本书将回顾云计算的发展情况，并讨论云计算安全中的概念、问题和挑战。此外，还将讨论我们在得克萨斯大学达拉斯分校（The University of Texas at Dallas，UTD）开发的在云计算和云计算安全方面的实验系统、基础设施和教育项目。

本书组织结构

本书共分为9个部分，每个部分分别描述和云计算开发与安全相关的一些方面。本书重点将是云计算查询处理以及周边安全问题。此外，本书也将讨论利用云计算的应用安全性，例如恶意软件检测、信息共享安全、内部威胁检测等。

本书第1章为引言。

第一部分由3章（第2~4章）组成，主要介绍云计算的支撑技术。这部分包括第2章所讨论的计算系统的演进，第3章讨论的安全技术，第4章讨论的数据、信息以及知识的管理技术。

第二部分由3章（第5~7章）组成，讨论服务和安全服务技术中的概念。其中，第5章讨论SOA（Service Oriented Architecture，面向服务的架构）和Web服务，面向服务的设计与分析以及安全服务。第6章将讨论语义Web服务和相关安全问题，在第7章将讨论专用的Web服务，如信息和知识管理服务。

第三部分由5章（第8~12章）组成，将讨论云计算中的核心课题。在第8章中，将讨论云计算的基本概念以及我们实现的一个云计算架构。在第9章，将对云计算功能进行讨论，包括虚拟化、数据存储管理、数据管理。接下来将用整个第10章来介绍云数据管理，针对云数据管理开发了许多原型。第11章讨论应用程序。第12章讨论各种云产品和安全提供商。

第四部分由3章（第13~15章）组成，讨论在实验云计算系统上开发的原型。首先将在第13章讨论云查询处理系统原型，其次将在第14章讨论云计算中的社交网络，最后在第15章讨论云计算中的本体管理。

第五部分由6章（第16~21章）组成，讨论云计算安全。第16章讨论云计算的安全性。第17章讨论云计算安全的主要功能，例如安全虚拟化、存储和数据管理以及云取证。由于我们大部分工作都是集中在云数据安全管理上，所以将用整个第18章来讨论这个课题。第19章给出了云计算安全指南。第20章讨论安全作为服务的概念。第21章讨论关于云计算安全的各种标准和产品。

第六部分由3章（第22~24章）组成，讨论在云计算上开发系统的安全性。第22章讨论的主题是使用关系数据的安全云查询处理系统，第23章将讨论使用语义Web数据的查询处理安全，第24章将讨论亚马逊网络服务和信息集成的安全性。

第七部分由4章（第25~28章）组成，介绍云计算在安全应用中的使用。第25章介绍使用云计算对恶意软件检测，第26章将讨论基于云的内部威胁检测，第27章讨论基于云计算的信息共享保障，第28章讨论基于语义云的信息共享，我们将语义云定义为使用语义Web技术的云。

第八部分由3章（第29~31章）组成，描述了关于开发可信赖云的思路。第29章将讨论云服务的信任管理，第30章对云服务的隐私问题进行讨论，第31章将对云服务的完整性和数据质量进行讨论。

第九部分由4章（第32~35章）组成，讨论了为云计算安全开发的基础设施、

教育项目和研究计划。第32章将介绍建立的云计算安全基础设施，第33章讨论关于这一主题的教育项目，第34章讨论了影响本书大部分讨论的协作研究项目。第35章对本书进行了总结，并讨论了云计算安全的未来发展方向。

每一部分的内容都是以简介开始，并以总结结尾。而且，从第2~34章每一章都是以概述开始，并以总结和对未来工作的展望结尾。此外，在每一章的结尾均提供了相关参考文献。我们给出了相关背景知识的4个附录。附录A是对数据管理的概述，并讨论了我们所撰写的书目之间的关系，这也是我们所有书籍的标准做法。附录B中，讨论了数据挖掘技术，因为我们使用这种技术用于云数据管理和基于云的恶意软件检测。附录C讨论了与提出的安全云数据管理工作相关的安全数据管理技术。最后，附录D讨论了在信息共享安全方面所做的工作，它与我们对基于云计算的信息共享安全相关。

编者尽力从各类书籍、期刊、杂志、会议和研讨会中获得参考。尽管编者试着不给出统一资源定位符（Uniform Resource Locator，URL）作为参考，但是发现这样几乎不可能完成一篇文章，特别是在关于云计算和网络方面。许多网址包含很多优质的参考资料，但其中一些网址在本书出版之前可能无法使用。因此，编者建议读者应经常检查网页以获取关于云计算和云计算安全的最新信息。

数据、信息和知识

通常，数据管理包括数据库管理、数据互操作性、数据迁移、数据仓储和数据挖掘。例如，为了提取信息和模式以及趋势，必须对网络上的数据进行管理和挖掘。数据可以存储在文件、关系数据库或其他类型的数据库中，例如多媒体数据库。数据可以是结构化的，也可以是非结构化的。本书将多次使用术语数据、数据管理、数据库系统和数据库管理系统。我们在附录中详细说明了这些术语。我们将数据管理系统定义为管理数据的系统，即从数据中提取有意义的信息并加以利用，因此，数据管理系统包括数据库系统、数据仓库以及数据挖掘系统。数据可以是关系数据库中的结构化数据，也可以是非结构化的数据，例如文本、语音、图像和视频等。

过去有很多关于区分数据、信息和知识的讨论。在我们之前一些关于数据管理和挖掘的书籍中，我们没有试图区分这些术语，我们只是简单地认为，数据可能只是一些位和字节，或者它可以向用户传达一些有意义的信息。然而，随着网络的出现，以及对数据、信息和知识管理作为不同的独立领域进行说明的需求越来越大，在本书中，对数据、信息和知识通过采取不同的方法尽可能来区分这些术语。我们认为数据通常是一些数值，如数字、整数和字符串。一些含义或语义与数据相关联时就会产生信息，例如约翰的工资是2万美元。知识是通过阅读和学习获得的，从而了解数据和信息并采取进一步的行为。也就是说，当人们消除了对数据和信息的不能理解的内容时，数据和信息就转化为知识。应该注意的是，对数据、信息和知识做到严格定义是相当困难的，有时我们也会互换使用这些术语。在附录中讨论的

数据管理相关内容有助于阐明一些差异。为了与我们之前书籍中的术语一致，我们还将区分数据库系统和数据库管理系统。数据库管理系统是管理包含持久性数据的数据库的软件，而数据库系统包含数据库和数据库管理系统。

结语

写作本书的目的是探索云计算的安全问题，并讨论其应用，同时也对云计算中的一些概念进行了讨论。我们讨论了安全的 Web 服务，因为它们是云计算的核心。除了一些概念和技术，还介绍了我们在 UTD 设计和开发的云计算实验系统和云计算安全实验系统。我们已经使用了这本书中的材料，以及每章中列出的众多参考文献，在 UTD 开设了 Web 安全服务和云计算方面的研究生课程。书中还提供了研究生开发的几个实验系统。

值得注意的是，云计算及安全领域正在随着新兴标准的出现不断迅速扩大。因此，读者需要跟上云计算和云计算安全的原型、产品、工具和标准的发展，这显得很重要。安全不能是在事后考虑的事情。所以，在云计算技术开发过程中，在一开始就要把安全性考虑进去是非常重要的。

关 于 作 者

Bhavani Thuraisingham 博士是得克萨斯大学达拉斯分校（UTD）埃里克乔森工程与计算机科学学院 Louis A. Beecherl，Jr. 杰出教授，以及 UTD 网络安全研究和教育研究所执行董事。她目前从事的是网络安全、云计算和大数据分析的集成法研究。在 2004 年 10 月加入 UTD 之前，她在 MITRE 公司工作了 16 年，其中担任了 3 年国家科学基金会（National Science Foundation，NSF）项目主任。在 MITRE 公司中，她领导团队努力开发安全数据管理系统和数据挖掘工具。她在 NSF 发起了数据和应用安全计划，并成为网络信任主题研究的一部分。在进入 MITRE 公司之前，她在商业行业中工作了 6 年，从事数据安全、数据管理和分布式系统的研究工作。

Thuraisingham 博士是许多奖项的获得者，包括 IEEE 计算机学会 1997 年技术成就奖、ACM SIGSAC 2010 年度杰出贡献奖、IEEE SMC 和社会交通系统 2010 年度智力与安全信息学领导联合奖、社会设计和过程科学（SDPS）2012 年度金融和计算机科学与社会科学跨学科研究金奖，以及 2013 年 IBM 学院奖。她是美国电气电子工程师学会（IEEE）、美国科学促进会（AAAS）和英国计算机学会的会士（Fellow）。她发表过 100 余篇期刊论文、200 余篇会议论文，出版过 12 本图书，完成超过 100 场主旨演讲。她已经拥有 8 项专利（3 项待定），并且是知识与安全分析有限责任公司的创始人。Thuraisingham 博士在威尔士大学获得了计算理论博士学位。由于在数据安全方面所做的贡献，她获得了英国布里斯托大学博士学位（工程博士）。

目　　录

译者序
原书前言
关于作者
第 1 章　引言…… 1
1.1　关于本书 …… 1
1.2　支撑技术 …… 2
1.2.1　从主机到云 …… 2
1.2.2　安全技术 …… 2
1.2.3　数据、信息和知识管理 …… 3
1.3　服务安全技术 …… 4
1.3.1　服务安全技术 …… 4
1.3.2　语义服务安全 …… 5
1.3.3　专用服务安全 …… 5
1.4　云计算概念 …… 6
1.5　云计算实验系统 …… 7
1.6　云计算安全 …… 7
1.7　云计算安全实验系统 …… 8
1.8　面向安全应用的云计算实验 …… 8
1.9　迈向可信赖的云 …… 9
1.10　面向云安全建立基础设施、教育项目和研究计划…… 9
1.11　本书的结构 …… 10
1.12　后续 …… 13
参考文献 …… 14
第一部分　支撑技术 …… 15
第 2 章　从主机到云…… 16
2.1　概述…… 16
2.2　早期计算系统…… 16
2.3　分布式计算…… 17
2.4　万维网…… 18
2.5　云计算…… 18
2.6　总结和展望…… 19
参考文献 …… 19
第 3 章　可信赖的系统 …… 20
3.1　概述…… 20
3.2　安全系统…… 20
3.2.1　简介…… 20
3.2.2　访问控制和其他安全概念…… 21
3.2.3　安全系统的类型…… 22
3.2.4　安全操作系统…… 23
3.2.5　安全数据库系统…… 23
3.2.6　安全的网络…… 24
3.2.7　新兴趋势…… 25
3.2.8　网络的影响 …… 26
3.2.9　构建安全系统的步骤…… 26
3.3　可靠的系统…… 27
3.3.1　简介…… 27
3.3.2　信任管理…… 28
3.3.3　数字版权管理…… 28
3.3.4　隐私…… 29
3.3.5　完整性、数据质量和高度保障…… 29
3.4　安全威胁和解决方案…… 30
3.5　通过不可信赖的组件构建安全系统…… 32
3.6　总结和展望…… 33
参考文献 …… 33
第 4 章　数据、信息和知识管理 …… 35
4.1　概述…… 35
4.2　数据管理…… 35
4.2.1　数据管理…… 35
4.2.1.1　数据模型…… 35
4.2.1.2　功能 …… 36
4.2.1.3　数据分布 …… 36
4.2.1.4　Web 数据管理…… 36
4.2.2　复杂数据管理…… 37

4.2.2.1　多媒体数据系统………… 37
4.2.2.2　地理空间数据管理……… 38
4.3　信息管理……………………………… 38
4.3.1　数据仓库和数据挖掘………… 39
4.3.2　信息检索……………………… 40
4.3.3　搜索引擎……………………… 41
4.4　知识管理……………………………… 42
4.5　活动管理……………………………… 43
4.5.1　电子商务和电子贸易………… 43
4.5.2　协作与工作流………………… 43
4.5.3　信息集成……………………… 45
4.5.4　信息共享……………………… 45
4.5.5　社交网络……………………… 46
4.5.6　供应链管理…………………… 47
4.6　总结和展望…………………………… 47
参考文献 ……………………………………… 48
第一部分总结 ………………………………… 49

第二部分　服务安全技术 ……… 50

第 5 章　面向服务的计算和安全 …… 51
5.1　概述…………………………………… 51
5.2　面向服务的计算 …………………… 52
5.2.1　服务模式……………………… 52
5.2.1.1　SOA 和 Web 服务 ……… 53
5.2.1.2　SOA 及其设计…………… 53
5.2.2　SOA 和 Web 服务 …………… 54
5.2.2.1　WS 模型 ………………… 55
5.2.2.2　WS 的构成 ……………… 56
5.2.2.3　WS 协议 ………………… 57
5.2.2.4　REST …………………… 59
5.2.3　SOAD ………………………… 59
5.2.3.1　IBM SOAD ……………… 61
5.2.3.2　面向服务的建模框架…… 62
5.2.3.3　服务型 UML …………… 62
5.3　面向服务计算的安全………………… 62
5.3.1　服务范式的安全……………… 62
5.3.2　安全的 SOA 和 WS ………… 65
5.3.2.1　WS - Security ………… 65
5.3.2.2　WS - * Security ……… 66
5.3.3　安全的 SOAD ………………… 69
5.3.3.1　安全的 SOMA …………… 71
5.3.3.2　安全的 SOMF …………… 72
5.3.3.3　用于服务的安全 UML … 72
5.3.4　WS 访问控制 ………………… 72
5.3.4.1　SAML …………………… 72
5.3.4.2　可扩展访问控制标记语言……………… 73
5.3.5　数字身份管理………………… 75
5.3.5.1　OpenID ………………… 76
5.3.5.2　Shibboleth ……………… 76
5.3.5.3　自由联盟………………… 78
5.3.6　WS 安全模型 ………………… 78
5.3.6.1　授权模型………………… 79
5.3.6.2　信息流模型……………… 79
5.3.6.3　多级安全 WS …………… 81
5.4　总结和展望…………………………… 81
参考文献 ……………………………………… 82

第 6 章　语义 Web 服务和安全 …… 85
6.1　概述…………………………………… 85
6.2　语义 Web …………………………… 86
6.2.1　分层技术栈…………………… 86
6.2.2　XML ………………………… 87
6.2.2.1　XML 语句和元素 ……… 87
6.2.2.2　XML 属性 ……………… 87
6.2.2.3　XML DTD ……………… 87
6.2.2.4　XML 模式 ……………… 87
6.2.2.5　XML 命名空间 ………… 88
6.2.2.6　XML 联合/分开 ………… 88
6.2.2.7　XML - QL、XQuery、XPath、XSLT ……………………… 88
6.2.3　RDF ………………………… 88
6.2.3.1　RDF 基础 ……………… 89
6.2.3.2　RDF 容器模型 ………… 89
6.2.3.3　RDF 规范 ……………… 89
6.2.3.4　RDF 模式 ……………… 90
6.2.3.5　RDF 公理语义学 ……… 90
6.2.3.6　RDF 推断法 …………… 90
6.2.3.7　RDF 查询 ……………… 91

6.2.3.8 SPARQL …… 91
6.2.4 本体 …… 91
6.2.5 Web 规则 和 SWRL …… 92
6.2.5.1 Web 规则 …… 92
6.2.5.2 SWRL …… 93
6.2.6 语义 Web 服务 …… 94
6.3 安全的语义 Web 服务 …… 95
6.3.1 语义 Web 的安全性 …… 95
6.3.2 XML 安全性 …… 96
6.3.3 RDF 安全性 …… 96
6.3.4 安全性和本体 …… 97
6.3.5 安全查询和规则处理 …… 97
6.3.6 语义 Web 的隐私和信任 …… 97
6.3.7 安全的语义 Web 和 WS …… 99
6.4 总结和展望 …… 100
参考文献 …… 101
第 7 章 专用的 Web 服务和安全 …… 103
7.1 概述 …… 103
7.2 专用 Web 服务 …… 103
7.2.1 总述 …… 103
7.2.2 数据管理的 Web 服务 …… 104
7.2.3 复杂的数据管理 Web 服务 …… 104
7.2.4 信息管理的 Web 服务 …… 106
7.2.5 知识管理的 Web 服务 …… 106
7.2.6 活动管理的 Web 服务 …… 107
7.2.6.1 电子商务 …… 107
7.2.6.2 协作与工作流 …… 108
7.2.6.3 信息集成 …… 108
7.2.6.4 其他活动 …… 109
7.2.7 领域 Web 服务 …… 109
7.2.7.1 防御 …… 109
7.2.7.2 医疗保健与生命科学 …… 110
7.2.7.3 金融 …… 110
7.2.7.4 电信 …… 111
7.2.8 新兴的 Web 服务 …… 111
7.2.8.1 X 即服务 …… 111
7.2.8.2 数据即服务 …… 111
7.2.8.3 软件即服务 …… 112
7.2.8.4 其他 X 即服务 …… 113
7.3 专用 Web 服务安全 …… 114
7.3.1 总述 …… 114
7.3.2 用于数据管理的 Web 服务安全 …… 114
7.3.3 用于复杂数据管理安全的 Web 服务 …… 115
7.3.3.1 地理空间数据管理安全 …… 115
7.3.3.2 安全的多媒体数据管理 …… 117
7.3.4 信息管理的安全 Web 服务 …… 118
7.3.5 知识管理的安全 Web 服务 …… 118
7.3.6 活动管理的安全 Web 服务 …… 119
7.3.6.1 安全电子商务 …… 119
7.3.6.2 供应链管理安全 …… 120
7.3.6.3 安全工作流和协作 …… 120
7.3.7 安全的领域 Web 服务 …… 123
7.3.7.1 防御 …… 123
7.3.7.2 医疗保健和生命周期 …… 124
7.3.7.3 金融 …… 124
7.3.7.4 其他领域 …… 124
7.3.8 新兴 Web 技术的安全 …… 124
7.3.8.1 X 即服务的安全性 …… 124
7.3.8.2 亚马逊 Web 服务的安全性 …… 125
7.3.8.3 为云和网格提供安全的 Web 服务 …… 125
7.4 总结和展望 …… 126
参考文献 …… 126
第二部分总结 …… 129
第三部分 云计算概念 …… 130
第 8 章 云计算概念 …… 131
8.1 概述 …… 131
8.2 云计算初步知识 …… 131
8.2.1 云的部署模型 …… 132
8.2.2 服务模型 …… 132
8.3 虚拟化 …… 133

8.4　云存储和数据管理 …… 133
8.5　总结和展望 …… 134
参考文献 …… 135
第9章　云计算功能 …… 136
9.1　概述 …… 136
9.2　云计算框架 …… 136
9.3　云操作系统和管理程序 …… 137
9.4　云网络 …… 138
9.5　云数据和存储管理 …… 139
9.6　云应用 …… 140
9.7　云策略管理、备份和恢复 …… 140
9.8　总结和展望 …… 141
参考文献 …… 141
第10章　云数据管理 …… 142
10.1　概述 …… 142
10.2　关系数据模型 …… 143
10.3　体系结构问题 …… 144
10.4　DBMS功能 …… 145
10.4.1　简介 …… 145
10.4.2　查询处理 …… 146
10.4.3　事务管理 …… 147
10.4.4　存储管理 …… 148
10.4.5　元数据管理 …… 149
10.4.6　数据库完整性 …… 150
10.4.7　容错 …… 151
10.5　数据挖掘 …… 151
10.6　其他方面 …… 153
10.7　总结和展望 …… 154
参考文献 …… 154
第11章　专用云、服务和应用 …… 155
11.1　概述 …… 155
11.2　专用云 …… 155
11.2.1　移动云 …… 155
11.2.2　多媒体云 …… 156
11.3　云应用 …… 157
11.4　总结和展望 …… 158
参考文献 …… 158
第12章　云服务提供商、产品和框架 …… 159
12.1　概述 …… 159
12.2　云服务提供商、产品和框架 …… 160
12.2.1　云服务提供商 …… 160
12.2.1.1　Windows Azure …… 161
12.2.1.2　谷歌应用引擎 …… 162
12.2.2　云产品 …… 162
12.2.2.1　Oracle企业管理器 …… 162
12.2.2.2　IBM智能云 …… 163
12.2.2.3　虚拟机管理程序产品 …… 164
12.2.3　云框架 …… 164
12.2.3.1　Hadoop、Map/Reduce框架 …… 164
12.2.3.2　Storm …… 165
12.2.3.3　Hive …… 166
12.3　总结和展望 …… 166
参考文献 …… 167
第三部分总结 …… 168
第四部分　云计算实验系统 …… 169
第13章　云查询处理实验系统 …… 170
13.1　概述 …… 170
13.2　我们的方法 …… 171
13.3　相关工作 …… 173
13.4　架构 …… 175
13.4.1　数据生成和存储 …… 176
13.4.2　文件组织 …… 176
13.4.3　谓词拆分 …… 176
13.4.4　使用客体的显式类型信息拆分 …… 177
13.4.5　使用客体的隐式类型信息拆分 …… 177
13.5　Map/Reduce框架 …… 177
13.5.1　简介 …… 177
13.5.2　输入文件选择 …… 178
13.5.3　查询处理的成本评估 …… 179
13.5.3.1　理想模型 …… 180
13.5.3.2　启发式模型 …… 181
13.5.4　查询计划生成 …… 182
13.5.4.1　最佳方案的计算复

杂度…………………… 184
13.5.4.2 条件放宽的最佳方案问题和近似解决方案………… 184
13.5.5 通过汇总统计分析突破限制………………………… 185
13.5.6 Map/Reduce 连接执行 …… 186
13.6 结果……………………………… 187
13.6.1 数据集、框架和实验配置 … 187
13.6.1.1 数据集………………… 187
13.6.1.2 基准框架……………… 188
13.6.1.3 实验配置……………… 188
13.6.2 评估……………………………… 188
13.7 总结和展望……………………… 190
参考文献 ……………………………… 190
第 14 章 云端社交网络 ………… 194
14.1 概述……………………………… 194
14.2 SNODSOC 和 SNODSOC + + 的基础技术………………………… 195
14.2.1 SNOD ……………………… 195
14.2.2 位置提取…………………… 196
14.2.3 实体/概念提取和集成 …… 197
14.2.3.1 语言扩展……………… 197
14.2.3.2 语言以外的扩展……… 197
14.2.3.3 实体整合……………… 197
14.2.4 本体构建…………………… 197
14.2.5 云查询处理………………… 198
14.2.5.1 预处理………………… 198
14.2.5.2 查询执行和优化……… 198
14.3 SNODSOC 设计 ……………… 198
14.3.1 模块概述…………………… 198
14.3.2 SNODSOC 和趋势分析 …… 199
14.3.2.1 新类检测……………… 200
14.3.2.2 存储簇汇总信息……… 200
14.3.3 内容驱动的位置提取……… 202
14.3.3.1 动机…………………… 203
14.3.3.2 挑战：拟议方案……… 204
14.3.3.3 使用地名词典和自然语言处理…………………… 206
14.3.4 分类………………………… 208
14.3.5 本体构建…………………… 209
14.4 走向 SNODSOC + + …………… 210
14.4.1 SNODSOC + + 的益处 …… 211
14.5 基于云的社交网络分析………… 211
14.5.1 流处理……………………… 212
14.5.2 SNODSOC 的 Twitter 风暴 … 212
14.6 相关工作………………………… 213
14.7 总结和展望……………………… 213
参考文献 ……………………………… 214
第 15 章 基于语义 Web 的云计算实验系统 ……………………… 217
15.1 概述……………………………… 217
15.2 Jena-HBase：分布式、可扩展和高效的 RDF 三元组存储 ……… 218
15.3 StormRider：利用社会网络的“Storm” ……………………… 219
15.4 使用 Map/Reduce 框架进行本体驱动的查询扩展………………… 221
15.4.1 使用 Map/Reduce 分布式方法对 BET 计算…………………… 222
15.4.1.1 使用迭代 Map/Reduce 算法的最短路径计算…… 223
15.4.1.2 使用 Map/Reduce 计算的中间性和中心度测度…………………… 224
15.4.1.3 使用 Map/Reduce 算法的 SSM ……………………… 225
15.5 总结和展望……………………… 225
参考文献 ……………………………… 225
第四部分总结 ……………………… 227
第五部分 云计算安全概念 … 228
第 16 章 云计算安全概念 ………… 229
16.1 概述……………………………… 229
16.2 云计算的安全和管理…………… 230
16.3 安全体系结构…………………… 231
16.4 身份管理和访问控制…………… 232
16.4.1 云身份管理………………… 233
16.5 云存储和数据安全……………… 234

16.6　云的隐私、合规性和取证……… 235
16.6.1　隐私……………………………… 235
16.6.2　条例和合规性………………… 235
16.6.3　云取证………………………… 236
16.7　加密方案…………………………… 236
16.8　网络安全…………………………… 237
16.9　业务连续性规划………………… 237
16.10　操作管理 ……………………… 238
16.11　物理安全 ……………………… 238
16.12　总结和展望 …………………… 239
参考文献 ………………………………… 239
第 17 章　云计算功能的安全 ……… 240
17.1　概述………………………………… 240
17.2　云计算安全框架………………… 240
17.3　云操作系统和管理程序安全…… 241
17.4　云网络安全……………………… 242
17.5　云存储管理安全………………… 243
17.6　云数据管理安全………………… 243
17.7　云安全和完整性管理…………… 244
17.8　云应用安全……………………… 244
17.9　总结和展望……………………… 245
参考文献 ………………………………… 245
第 18 章　云数据管理安全 ………… 246
18.1　概述………………………………… 246
18.2　数据管理安全…………………… 246
18.2.1　访问控制……………………… 246
18.2.2　推理问题……………………… 247
18.2.3　分布式/异构数据管理安全 … 248
18.2.4　对象数据系统安全…………… 248
18.2.5　数据仓库、数据挖掘、安全和隐私…………………… 249
18.2.6　信息管理安全………………… 249
18.2.7　知识管理安全………………… 250
18.3　云的影响………………………… 251
18.3.1　自主安全……………………… 251
18.3.2　推理问题……………………… 251
18.3.3　分布式和异构数据管理安全……………………………… 251
18.3.4　对象系统安全………………… 252
18.3.5　数据仓库、数据挖掘、安全和隐私………………………… 252
18.3.6　信息管理安全………………… 252
18.3.7　知识管理安全………………… 252
18.4　总结和展望……………………… 252
参考文献 ………………………………… 253
第 19 章　云计算安全指南 ………… 254
19.1　概述………………………………… 254
19.2　指南………………………………… 254
19.3　总结和展望……………………… 257
参考文献 ………………………………… 257
第 20 章　安全即服务 ……………… 258
20.1　概述………………………………… 258
20.2　用于网络安全的数据挖掘服务应用………………………… 259
20.2.1　简介 ………………………… 259
20.2.2　网络恐怖主义、内部威胁和外部攻击 ………………… 259
20.2.3　恶意入侵……………………… 260
20.2.4　信用卡欺诈和身份盗用…… 260
20.2.5　对关键基础设施的攻击…… 260
20.2.6　用于网络安全的数据挖掘服务 ………………………… 261
20.3　目前关于安全即服务的研究…… 262
20.4　网络安全应用程序的其他服务 … 263
20.5　总结和展望……………………… 263
参考文献 ………………………………… 264
第 21 章　云计算安全产品 ………… 265
21.1　概述………………………………… 265
21.2　产品概述………………………… 265
21.3　总结和展望……………………… 268
参考文献 ………………………………… 268
第五部分总结 …………………………… 270
第六部分　云计算安全实验系统 …… 271
第 22 章　对关系数据进行安全的云查询处理 ………………………… 272
22.1　概述………………………………… 272

22.2 相关工作 …… 273
22.3 系统架构 …… 274
22.3.1 Web 应用层 …… 274
22.3.2 ZQL 解析器层 …… 274
22.3.3 XACML 策略层 …… 276
22.3.3.1 XACML 策略构建器 …… 276
22.3.3.2 XACML 策略评估器 …… 276
22.3.3.3 基本查询重写层 …… 276
22.3.3.4 Hive 层 …… 277
22.3.3.5 HDFS 层 …… 277
22.4 实验细节和结果 …… 277
22.4.1 实验配置 …… 277
22.4.2 实验数据集 …… 278
22.4.3 实验结果 …… 278
22.5 总结和展望 …… 279
参考文献 …… 280
第 23 章 对语义 Web 数据的云查询处理安全 …… 282
23.1 概述 …… 282
23.2 背景 …… 283
23.2.1 相关工作 …… 283
23.2.1.1 Hadoop 和 Map/Reduce …… 284
23.3 访问控制 …… 284
23.3.1 模型 …… 284
23.3.2 AT 任务 …… 286
23.3.2.1 代理 AT 的最终输出 …… 286
23.3.2.2 安全级别默认值 …… 286
23.3.3 冲突 …… 287
23.4 系统架构 …… 288
23.4.1 系统架构概述 …… 288
23.4.1.1 数据生成和存储 …… 288
23.4.1.2 示例数据 …… 289
23.5 策略实施 …… 290
23.5.1 查询重写 …… 290
23.5.2 嵌入式执行 …… 290
23.5.3 后处理执行 …… 292
23.6 实验配置和结果 …… 292
23.7 总结和展望 …… 292
参考文献 …… 293
第 24 章 基于云的信息集成安全 …… 295
24.1 概述 …… 295
24.2 将 Blackbook 与 Amazon S3 集成 …… 296
24.3 实验 …… 300
24.4 总结和展望 …… 300
参考文献 …… 301
第六部分总结 …… 302
第七部分 面向安全应用的云实验系统 …… 303
第 25 章 面向演进数据流的基于云的恶意软件检测 …… 304
25.1 概述 …… 304
25.2 恶意软件检测 …… 305
25.2.1 作为数据流分类问题的恶意软件检测 …… 305
25.2.2 将云计算用于检测恶意软件 …… 306
25.2.3 我们的贡献 …… 307
25.3 相关工作 …… 307
25.4 系统设计与实现 …… 309
25.4.1 集成结构与更新 …… 309
25.4.2 误差减少分析 …… 310
25.4.3 经验性误差减少与时间复杂度 …… 311
25.4.4 Hadoop/Map/Reduce 框架 …… 311
25.5 恶意代码检测 …… 313
25.5.1 简介 …… 313
25.5.2 非分布式特征提取与选择 …… 313
25.5.3 分布式特征提取与选择 …… 314
25.6 实验 …… 315
25.6.1 数据集 …… 315
25.6.1.1 合成数据集 …… 315
25.6.1.2 僵尸网络数据集 …… 316
25.6.1.3 恶意软件数据集 …… 316
25.6.2 基线方法 …… 317
25.6.2.1 Hadoop 分布式系统

配置 …………………… 317
25.7 讨论 ………………………… 317
25.8 总结和展望 ………………… 318
参考文献 ………………………… 319
第 26 章 用于内部威胁检测的基于云计算的数据挖掘 ……… 322
26.1 概述 ………………………… 322
26.2 挑战、相关工作和我们的方法 … 322
26.3 用于内部威胁检测的数据挖掘 … 323
26.3.1 我们的解决方案的体系结构 ………………………… 323
26.3.2 特征提取和紧凑表示 ……… 324
26.3.2.1 子空间聚类 …………… 325
26.3.3 RDF 仓库架构 …………… 326
26.3.4 数据存储 ………………… 327
26.3.4.1 文件组织 ……………… 328
26.3.4.2 谓词分割 ……………… 328
26.3.4.3 谓词客体分割 ………… 328
26.3.5 使用 Hadoop/Map/Reduce 技术应答查询 ………………… 328
26.3.6 数据挖掘应用 …………… 329
26.4 综合框架 …………………… 330
26.5 总结和展望 ………………… 331
参考文献 ………………………… 331
第 27 章 以云为中心保障信息共享 ………………………… 333
27.1 概述 ………………………… 333
27.2 系统设计 …………………… 335
27.2.1 CAISS 设计 ……………… 335
27.2.1.1 增强策略引擎 ………… 336
27.2.1.2 增强 SPARQL 查询处理器 ………………… 336
27.2.1.3 集成框架 ……………… 336
27.2.2 CAISS++的设计 ………… 338
27.2.2.1 CAISS 的局限性 ……… 338
27.2.2.2 CAISS++设计 ………… 338
27.2.2.3 集中式 CAISS++ …… 339
27.2.2.4 分散式 CAISS++ …… 341
27.2.2.5 混合 CAISS++ ……… 342
27.2.2.6 命名规则 ……………… 345
27.2.2.7 垂直分区布局 ………… 345
27.2.2.8 混合布局 ……………… 345
27.2.2.9 SPARQL 的分布式处理 ………………… 346
27.2.2.10 框架集成 …………… 346
27.2.2.11 策略规范和实施 …… 347
27.2.3 形式策略分析 …………… 347
27.2.4 实施方法 ………………… 348
27.3 相关工作 …………………… 348
27.3.1 我们的相关研究 ………… 348
27.3.1.1 云中的数据存储和检索安全 ………………… 348
27.3.1.2 云上的 SPARQL 查询处理安全 ………………… 349
27.3.1.3 RDF 策略引擎 ………… 350
27.3.1.4 ASI 原型 ……………… 351
27.3.1.5 形式策略分析 ………… 351
27.3.2 总体相关研究 …………… 352
27.3.2.1 云中的数据存储和检索安全 ………………… 352
27.3.2.2 SPARQL 查询处理器 ………………… 352
27.3.2.3 基于 RDF 的策略引擎 ………………… 352
27.3.2.4 Hadoop 存储架构 …… 353
27.3.2.5 分布式推理 …………… 353
27.3.2.6 访问控制和策略本体建模 ………………… 353
27.3.3 商业发展 ………………… 353
27.3.3.1 RDF 处理引擎 ………… 353
27.3.3.2 基于 Web 的语义安全策略引擎 ………………… 354
27.3.3.3 云 …………………… 354
27.4 总结和展望 ………………… 354
参考文献 ………………………… 354
第 28 章 基于语义云信息共享保障系统的设计与实现 ……… 358
28.1 概述 ………………………… 358

28.2 架构…… 358
28.2.1 简介…… 358
28.2.2 框架配置…… 359
28.2.3 架构中的模块…… 360
28.2.3.1 用户接口层…… 360
28.2.3.2 策略引擎…… 361
28.2.3.3 数据层…… 366
28.2.4 我们策略引擎框架的特点…… 366
28.2.4.1 策略互惠…… 367
28.2.4.2 条件策略…… 367
28.2.4.3 策略对称性…… 367
28.2.4.4 制定和衡量策略…… 367
28.2.4.5 合理利用资源…… 368
28.2.4.6 策略规范和执行…… 368
28.3 总结和展望…… 368
参考文献 …… 369
第七部分总结 …… 370

第八部分 迈向可信赖的云 … 371

第29章 信任管理和云计算 …… 372
29.1 概述…… 372
29.2 信任管理…… 373
29.2.1 信任管理与协商…… 373
29.2.2 信任与风险管理…… 374
29.2.3 基于声誉的系统…… 375
29.3 信任和云服务…… 375
29.3.1 信任管理即云服务…… 375
29.3.2 云服务的信任管理…… 377
29.4 总结和展望…… 378
参考文献 …… 378

第30章 隐私和云服务 …… 379
30.1 概述…… 379
30.2 隐私管理…… 379
30.2.1 隐私问题…… 379
30.2.2 推理产生的隐私问题…… 380
30.2.3 隐私偏好平台…… 381
30.2.4 保护隐私的云挖掘…… 382
30.3 隐私管理和云…… 383
30.3.1 云服务的隐私管理…… 383
30.3.2 云服务和语义云服务中的隐私…… 384
30.4 总结和展望…… 385
参考文献…… 385

第31章 完整性管理、数据来源和云服务 …… 386
31.1 概述…… 386
31.2 完整性、数据质量和数据来源…… 386
31.2.1 完整性方面…… 386
31.2.2 推论、数据质量和数据来源…… 387
31.3 完整性管理和云服务…… 388
31.3.1 用于完整性管理的云服务…… 388
31.3.2 云和语义云服务的完整性…… 390
31.4 总结和展望…… 391
参考文献 …… 391
第八部分总结 …… 392

第九部分 面向云安全建立基础设施、教育项目和研究计划 … 393

第32章 云安全的基础设施 …… 394
32.1 概述…… 394
32.2 基础设施研究描述…… 395
32.2.1 背景…… 395
32.2.1.1 我们基础设施的需求 … 395
32.2.1.2 云计算中的 Hadoop … 396
32.2.1.3 Hadoop 的不足 …… 396
32.2.2 基础设施开发…… 397
32.2.3 基础设施的硬件组件…… 398
32.2.3.1 硬件组件的集群部分 … 398
32.2.3.2 硬件组件中的安全协处理器部分…… 399
32.2.4 基础设施的软件组件…… 400
32.2.4.1 用于存储、查询和挖掘语义

Web 数据的组件部分 … 400
32.2.4.2 使用 IRM 将 SUN XACML 实现集成到 HDFS 中 … 401
32.2.4.3 强认证的组件部分…… 403
32.2.5 基础设施的数据组件……… 403
32.3 将云与现有基础设施相集成…… 403
32.4 利用云基础设施的示例项目…… 404
32.5 教育与实施……………………… 405
32.5.1 加强教育…………………… 405
32.5.2 实施………………………… 406
32.6 总结和展望……………………… 406
参考文献 ………………………… 406

第 33 章　云安全的教育项目 ……… 409
33.1 概述……………………………… 409
33.2 UTD 的信息保障教育 ………… 410
33.2.1 UTD CS 概述 ……………… 410
33.2.2 IA 提供的课程 …………… 411
33.2.3 我们关于 IA 的教育项目 … 412
33.2.4 IA 教育与研究的设备和设施………………………… 412
33.3 云计算教育项目的保障措施…… 413
33.3.1 能力建设活动组织………… 413
33.3.2 课程开发活动……………… 414
33.3.2.1 顶点课程……………… 414
33.3.2.2 组件插入现有课程…… 416
33.3.3 课程编程项目……………… 417
33.3.3.1 为了存储安全的细粒度访问控制……………… 418
33.3.3.2 灵活认证……………… 418
33.3.3.3 虚拟机安全管理……… 418
33.3.3.4 云的安全协处理器（SCP） ……… 418
33.3.3.5 恶意代码检测的可扩展技术…………… 418
33.3.4 云计算教学设施…………… 419
33.4 评估计划………………………… 419
33.5 总结和展望……………………… 420
参考文献 ………………………… 420

第 34 章　云安全的研究计划 ……… 421
34.1 概述……………………………… 421
34.2 研究贡献………………………… 421
34.2.1 简介………………………… 421
34.2.2 云数据安全和信息管理…… 422
34.2.2.1 云中的数据密集查询处理安全………………… 422
34.2.2.2 混合云中的数据处理安全………………… 422
34.2.2.3 云中的信息集成安全 … 423
34.2.2.4 云中社交网络的安全 … 424
34.2.3 基于云的安全应用………… 424
34.2.3.1 面向演进数据流的基于云的恶意软件检测………… 424
34.2.3.2 基于云的内部威胁检测 ………………… 425
34.2.3.3 基于云的信息共享保障（AIS） ………… 425
34.2.4 云的安全模型……………… 425
34.2.4.1 服务云中的一种信息流控制的细粒度模型……… 425
34.2.4.2 CloudMask：细粒度属性访问控制……………… 426
34.2.4.3 存储即服务模型中的委托访问控制……………… 427
34.2.4.4 基于属性的组密钥管理方案……………… 427
34.2.4.5 云中的隐私保护访问控制……………… 427
34.2.5 在云中建立安全的社交网络……………………… 428
34.2.5.1 安全的社交网络……… 428
34.2.5.2 数据的可信度………… 429
34.2.5.3 文本挖掘与分析……… 429
34.3 总结和展望……………………… 429
参考文献 ………………………… 430

第 35 章　本书总结和展望 ………… 432
35.1 关于本章………………………… 432
35.2 本书总结………………………… 432
35.3 云计算和云计算安全的方向…… 434

35.3.1 安全的服务 …………………… 434
35.3.2 云计算 …………………… 435
35.3.3 云计算安全 …………………… 435
35.4 我们保护云的目标 …………………… 435
35.5 我们该怎么继续发展 …………………… 436
第九部分总结 …………………… 437

附录 …………………… 438

附录 A 数据管理系统——发展与趋势 …………………… 438
A.1 概述 …………………… 438
A.2 数据库系统的发展 …………………… 439
A.3 地位、愿景和问题 …………………… 442
A.4 数据管理系统框架 …………………… 442
A.5 构建基于框架的信息系统 …… 444
A.6 书籍之间的关系 …………………… 446
A.7 总结和展望 …………………… 447
参考文献 …………………… 448
附录 B 数据挖掘技术 …………………… 449
B.1 概述 …………………… 449
B.2 数据挖掘任务和技术概述 …… 449
B.3 人工神经网络 …………………… 450
B.4 支持向量机 …………………… 453
B.5 马尔可夫模型 …………………… 455
B.6 关联规则挖掘 …………………… 457
B.7 多类问题 …………………… 460
B.7.1 一对一 …………………… 460
B.7.2 一对全部 …………………… 460
B.8 图像挖掘 …………………… 460
B.8.1 特征选择 …………………… 461
B.8.2 自动图像注释 …………………… 461
B.8.3 图像分类 …………………… 462
B.9 总结和展望 …………………… 462
参考文献 …………………… 462
附录 C 数据库系统中的访问控制 …… 463
C.1 概述 …………………… 463
C.2 安全策略 …………………… 464
C.2.1 访问控制策略 …………………… 464
C.2.2 管理策略 …………………… 467
C.2.3 识别和认证 …………………… 468
C.2.4 审计数据库系统 …………………… 468
C.2.5 安全视图 …………………… 468
C.3 策略实施及相关问题 …………………… 468
C.3.1 为了安全的 SQL 扩展 …… 469
C.3.2 查询修改 …………………… 470
C.3.3 自主安全性和数据库功能 …………………… 471
C.3.4 策略的可视化 …………………… 471
C.3.5 原型和产品 …………………… 473
C.4 总结和展望 …………………… 473
参考文献 …………………… 474
附录 D 信息共享保障的生命周期 …… 475
D.1 概述 …………………… 475
D.2 问题 …………………… 475
D.3 信息共享保障的生命周期 …… 476
D.4 激励和信息共享 …………………… 477
D.5 总结和展望 …………………… 478
参考文献 …………………… 478

致谢 …………………… 479

第 1 章　引　　言

1.1　关于本书

19 世纪，查尔斯·巴贝奇发明了机械计算机器。20 世纪 30 年代，图灵证明了著名的停机问题是无法解决的。在随后的 20 世纪 40 年代，冯·诺依曼计算机的产生为现代电子计算机奠定了基础。紧接着是 50 年代和 60 年代的大型计算机、70 年代的小型计算机、80 年代的个人计算机、90 年代的移动/无线计算机和 21 世纪初的云计算。同时，计算模式也从单处理器发展到多处理器，再到分布式和并行处理。现在，我们已经有了在单芯片上集成多处理器的多核架构。

在过去的十年中，随着万维网（World Wide Web，WWW）的发展，消费者服务提供商的概念已经通过网络进行了数字化体现。通过这种方式，我们现在拥有了 Web 支持的服务。消费者可以通过服务提供商提供的网站来请求服务，而服务提供商提供所请求的服务。例如可以通过这种服务预订机票或者网购一本书。这种 Web 支持的服务被称为网络服务。支持服务实现的信息系统是面向服务的信息系统。一个提供支持服务实现的架构被称为面向服务的架构（SOA）。

经由云计算和面向服务计算的集成，现在可以通过云来提供服务。这些服务不仅是域服务，如预订旅游或酒店等，而且还包括通过云端进行整个计算，并将其作为服务提供给客户。那么什么是云？美国国家标准与技术研究院给出的定义如下：

云计算是实现无处不在、方便、按需网络访问共享的可配置计算资源池（如网络、服务器、存储、应用程序和服务）的模型，可以通过最少的管理努力或服务提供商的交互来快速地提供和发布。

本章详细介绍本书的组织结构。本章组织如下：1.2 节论述用于构建和保证云安全的支撑技术。1.3 节论述云计算安全的核心——服务安全技术。1.4 节论述云计算的概念。1.5 节论述我们已开发的云实验系统。1.6 节论述云安全技术。1.7 节论述我们已开发的云安全实验系统。1.8 节论述利用云来保障安全应用的实验系统。1.9 节论述一些构建可信赖的云的方向，包括机密性、信任性、隐私性和完整性。最后，在 1.10 节中论述我们在构建云安全基础架构、教育项目以及我们在云安全上的协作活动方面等做出的努力。1.11 节介绍本书的组织结构。我们在 1.12 节中提出用到的资源。需要注意的是，1.2 ~ 1.10 节的内容将在本书第一 ~ 九部分中进行详细阐述。图 1.1 简要说明了本章内容。

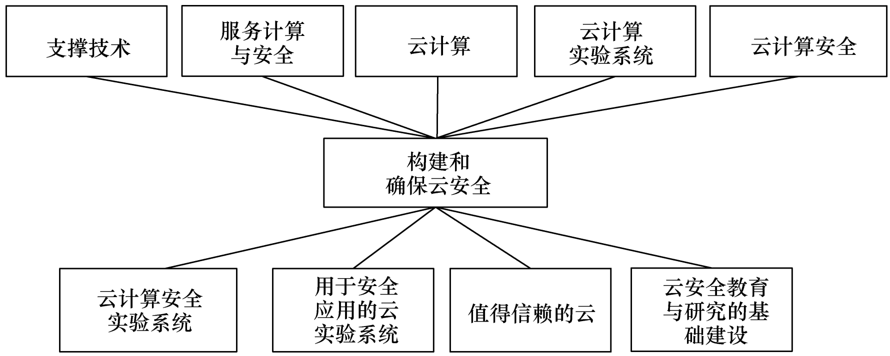


图 1.1　构建和确保云安全

1.2　支撑技术

本节将详述 3 种云的支撑技术。首先，将论述计算系统的演变。然后，将论述安全技术。最后，将论述数据、信息和知识管理技术。本书第二部分将论述支撑技术的更多细节。图 1.2 说明了支撑技术。

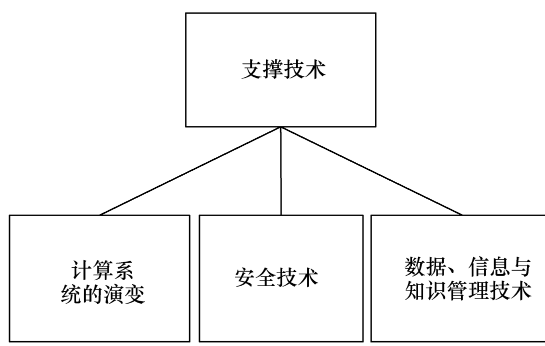


图 1.2　支撑技术

1.2.1　从主机到云

如 1.1 节所述，计算系统在过去的 70 年里持续发展。20 世纪 30 年代的结果不可解性与 40 年代冯・诺依曼计算机为现代计算机的出现铺平了道路。我们看到 50 年代和 60 年代的大型计算机、70 年代的小型计算机和 80 年代的个人计算机的快速发展。随着网络的发展，在 20 世纪 70 年代和 80 年代产生了分布式系统。

随着 WWW（万维网）的到来，新一代计算技术开始于 20 世纪 90 年代。这导致了语义 Web 与计算服务技术产生。在 21 世纪初出现了社会计算、移动计算和云计算。我们预计这 3 种计算技术将在未来集成为一体并拥有处理大量数据的能力，这被称为“大数据”。也就是说，计算的未来将是快速处理大量数据，并通过手机向用户提供有用的信息。本书将重点介绍通过云服务提供的数据管理、信息管理和社交网络的一些发展。有关计算演进的更多细节将在本书第 2 章中给出。

1.2.2　安全技术

安全的云计算系统本质上是云计算技术和安全技术的结合。安全技术将被纳入可靠

的信息系统。这些系统由许多方面组成，包括可信赖的系统、数据安全和信息系统，以及保密、隐私和信任管理方面，这些都将在本书中提到。

可信赖的系统是安全可靠的系统。至于可靠的系统，我们的意思是说具有高完整性、容错性和满足实时限制的系统。可信赖的系统包括数据管理系统、信息管理系统和安全的网络。要使一个系统可靠，它必须安全、容错、满足按时调度，并管理高质量的数据。然而，将这些功能集成到一个系统意味着系统必须满足处理冲突的要求。例如，如果系统对所有的访问进行控制检查，其中可能会产生超时现象。要设计可信赖的系统，最重要的是所设计系统的灵活性。例如，在某些情况下，满足所有的时限性约束可能很重要，而在某些其他情况下，满足所有安全约束可能至关重要。

可信赖的系统有时被称为可靠的系统，而在某些其他情况下，可靠性被认为是可信赖的一部分。例如，在一些论文中，可靠性主要包括容错系统，当将容错与安全集成在一起时，就可以得到值得信赖的系统。不管如何定义，对于要在操作环境中部署的系统，特别是对于命令和控制以及其他关键应用程序，我们需要端到端的可靠性和安全性。对于某些应用程序，我们不仅需要安全性和保密性，还需要确保个人的隐私安全。因此，隐私性也是可信赖系统的另一个特点。

为了使系统可靠/值得信赖，我们需要保证其端到端的可靠性或可信性。需要注意的是，组成系统的组件包括网络、操作系统、中间件和基础设施、数据管理器和应用程序。我们需要所有的组件是可靠的或可信赖的。然而在最近，安全系统的目标是通过不可靠的组件构建可信赖的系统。假设组件可能来自多个提供商，甚至来自多个国家，那么信任所有组件是不可行的。因此，这里的挑战在于如何由不可靠的组件构建可信赖的系统，并确保任务的正确执行。

安全服务的安全技术包括数据安全服务、信息和知识管理服务。安全数据和信息系统包括数据库系统安全（如关系数据库系统安全）和信息系统安全（如多媒体信息系统安全和数字图书馆安全）。

可信赖的信息系统的重点在于保密性、隐私性和可信任性的概念。保密性是指网站或服务器只对授权的个人发送有关的信息。隐私性是指个人决定发送其有关信息。可信任性是关于可以对各个人（例如，个人、组织）及其产生的信息有多少价值。我们将在第 3 章论述安全技术。

1.2.3 数据、信息和知识管理

云数据管理是本书的重点，由于我们专注于这一领域，所以认为数据、信息和知识管理是云的关键支撑技术。应该注意的是，包括操作系统、网络和中间件在内的几种计算技术也是支撑技术。

本书重点介绍的数据管理技术，包括查询处理、事务管理、存储管理以及数据挖掘和语义 Web 等信息管理技术。此外，我们将专注于在云端托管应用程序，如社交网络。知识管理也是一种可以通过云提供的技术。本书将在第 4 章论述数据、信息和知识管理的各个方面。

1.3 服务安全技术

服务是云计算的核心。这是因为云提供了一系列的服务（包括平台即服务、软件即服务、数据即服务、基础设施即服务）给它们的客户。因此，我们将把第二部分用于介绍服务和服务安全技术。本书将论述的各种类型的服务如图 1.3 所示。

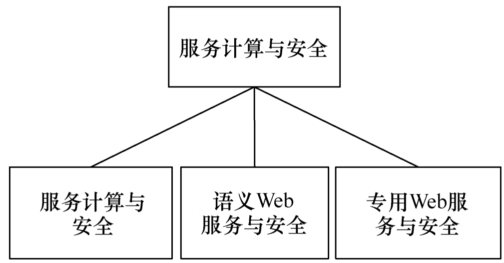


图 1.3 服务类型

1.3.1 服务安全技术

服务的基本技术由多个要点组成，包括面向服务的概念、SOA 和网络服务、语义 Web 服务和面向服务的分析与设计（SOAD）。面向服务的计算由面向对象的计算演变而来。在面向对象的计算中，世界被看作是对象的集合，这些对象通过消息相互通信。同样，在面向服务的计算中，世界被看作是服务的集合。因此，服务就可以像消费者那样通过信息交流。虽然众多的面向对象编程（Object – Oriented Programming，OOP）语言已被开发，包括 Smalltalk、Java 和 C + +，但现在仍没有面向服务计算的计算语言。然而，我们可以通过网络数据包和对象来实现服务。

SOA 是面向服务计算的基础架构。这种架构指定了服务和服务之间的通信。Web 服务是通过网络涉及的服务。目前，Web 服务是面向服务计算最流行的表达方式，但应该指出的是，Web 服务只是面向服务的计算的一个子集。

SOAD 遵循面向对象的分析和设计（OOAD）的规范。值得注意的是，OOAD 就如统一建模语言（Unified Modeling Language，UML）一样，已经过多年的研究和实验发展。这里的想法是开发一种方法为围绕对象概念的应用程序、数据和活动建模。同样，关于 SOAD，我们的想法是建立模型，进行分析，并寻找服务和服务之间交互的原理。通常，OOAD 被视作 SOA 的先决条件。尽管对于 OOP 来说 OOAD 是不可少的，但为了达成简单对象访问协议（Simple Object Access Protocol，SOAP），我们需要 SOAD。在本书的第二部分，将详细介绍所有的基本服务。

对于许多应用程序，云计算提供的服务必须是安全的。也就是说，不仅要保证数据存储和管理的安全，同时也要保证对数据的处理也是安全的。因此，对于云计算安全来讲，Web 服务安全是必不可少的。现在，基于 SOA 的 Web 服务由 3 个概念组成：消费者、服务提供商和目录。［这个目录被称为 UDDI（通用描述、发现和集成，Universal Description，Discovery and Integration）］。服务提供商将在 UDDI 上发布其服务，消费者将在 UDDI 上查询服务。UDDI 将给消费者提供服务提供商的地址。然后消费者调用服务。通过被称为 SOAP 的同步对象访问协议（Synchronized Object Access Protocol，SOAP）的消息进行通信，该消息基于可扩展标记语言（Extensible Markup Language，XML）。从安

全角度来看，我们需要将安全性纳入此通信。Web 服务 1.0 对服务安全提供了支持。在最近的 Web 服务 2.0 中，SOA 有了更加先进的概念。

在我们设计基于服务的安全应用之前，需要应用 SOAD 方法。然而，SOAD 就如同被定义的那样，不足以用于安全应用。在本书中，将论述我们为安全应用而拓展 SOAD 的工作。我们将基于在 20 世纪 90 年代进行的工作，扩展对象建模技术用于安全应用。

已经开发了针对服务的多项标准，安全的服务主要基于 Web 服务已经开发了多种标准。例如，万维网联盟（World Wide Web Consortium，W3C）已经开发了 XML 和 XML 安全标准，这是 Web 服务框架的重要组成部分。此外，诸如 WSDL（网络服务描述语言，Web Services Description Language）等标准已经被制定，这是为了指定用于消息通信的服务和 SOAP。XML 安全标准包括 XML 加密和 XML 密钥管理，这确保了机密性和完整性。

指定安全策略的更高级标准包括 OASIS（结构化信息标准促进组织，Organization for the Advancement of Structured Information Standards）以及 W3C 提出的标准。例如，安全声明标记语言（Security Assertions Markup Language，SAML）为安全性声明指定了一种语言。可扩展访问控制标记语言（Extensible Access Control Markup Language，XACML）指定了用于策略规范的语言。此外，还有联盟身份管理的标准，如自由联盟的联盟工作和高级 Web 服务安全性标准，包括 WS 和 WS * 安全性。我们对 Web 服务安全的一些研究也包括授权模型。

1.3.2 语义服务安全

语义服务是将服务技术与语义 Web 技术相结合的服务。语义 Web 由 Tim Berners Lee 发明，以支持网页可以被机器理解的想法。如今，语义 Web 被看作是一个通过与语义相关的节点和链接构成的非常大的链接图。语义 Web 是包括 XML 和资源描述框架（Resource Description Framework，RDF）在内的一系列技术的集合。这些利用语义 Web 技术的网络服务是语义 Web 服务。

语义信息系统安全本质上集成了语义 Web、服务和安全技术。语义 Web 技术可提供机器理解的网络服务。利用语义 Web 技术的网络服务可以处理语义异构性和其他互操作性问题。Web 服务安全需要使用语义 Web 安全技术。Web 服务所使用的 XML 和 RDF 文档必须是安全文档。语义 Web 服务安全的一些关键方面包括 XML 安全性和 Web 服务/RDF 安全性和 Web 服务，本体安全和 Web 服务，最后是规则安全和 Web 服务安全。我们定义，利用语义服务的云是语义云。我们的许多实验系统都要利用语义云。

1.3.3 专用服务安全

专用的网络服务包括安全的数据、信息、知识和活动管理服务以及域网络服务。数据管理服务包括那些用于事务安全、存储安全和查询处理安全的服务。复杂的数据管理服务包括安全的多媒体和地理空间网络服务。信息管理服务包括信息检索安全和信息安全可视化。知识管理服务包括知识产权管理安全。活动管理服务包括电子商务安全和信

息共享保障（Assured Information Sharing，AIS）。

我们将详细说明一些问题。知识管理是关于重复使用组织的知识和专长来提高利润和其他收益。在本书的第一部分和第二部分中，我们将研究更详细的数据、信息和知识管理的安全问题，然后讨论语义 Web 技术被应用于数据、信息和知识的管理中。异构数据源的互操作性是许多应用程序的关键。面临的挑战是如何在不同的安全系统中相互操作？如何整合异构的安全策略？电子商务（也被称为电子贸易）是指组织在网上进行交易。电子商务的各种模型、架构和技术正在被开发。由于我们正在处理的是像资金和账户等重要的数据，所以开展电子商务时，这些信息的保密性和隐私性是至关重要的。我们还必须确保数据没有恶意损坏。AIS 是共享信息的组织，但为了集成和挖掘数据去提取有用信息，且同时保持安全性，AIS 也要实施策略和程序。对于上述所有应用，网络服务均起着重要的作用。

人们正在为许多应用部署 Web 服务，包括医疗、金融、指挥和控制及电信。这也适用于诸如电子商务、知识管理，以及 AIS 等技术。其他新兴的 Web 服务包括用于网格的 Web 服务（最近诸如 Oracle 的公司正在开发基于网格的 Web 服务）。此外，亚马逊网络服务也基于网格范式。

其他类型的网络服务，其中包括数据即服务和软件即服务。例如，数据中心管理客户的数据。客户可以通过网络服务调用这样的数据服务。客户还可以调用各种软件（如编译器和操作系统）作为网络服务。在本书的第三部分和第五部分中，我们将论述为了云中的数据、信息、知识和活动管理，而使用的网络服务及其安全。

1.4 云计算概念

云计算是信息技术和数据处理领域中新兴的计算模式。企业利用云计算服务将数据维护外包，可以带来巨大的经济效益。企业从远程位置的“云端”存储和访问数据。新兴的云计算模式试图解决网络设备连接的增长的问题，并处理海量数据。谷歌现在推出了 Map/Reduce 框架，用于处理商品硬件上的大量数据。Apache 的 Hadoop 分布式文件系统（Apache’s Hadoop Distributed File System，HDFS），与 Map/Reduce 等集成部件组合，正在成为云计算的高级软件组件。

人类的推理、解释和决策能力需求的增长导致了语义 Web 的出现，这是一个积极的尝试：将网络从目前的只有人类可读的形式转变为机器可处理的形式。这反过来导致大量社交网站包含大量数据被共享和管理。因此，我们需要可扩展技术来处理大量的站点和处理大量的数据。

云计算技术将在本书的第三部分详述。其主题包括：①云计算功能，如虚拟化和存储管理；②云和数据的管理功能，如云查询处理和事务管理；③云计算应用；④云计算产品和工具。我们也论述了云计算框架。图 1.4 说明了云计算技术的相关主题。

请注意，云实质上是为客户提供一系列服务。这些服务包括基础设施服务、平台服务和应用服务，以及数据、信息、知识和活动管理服务。云也可以提供语义服务，我们

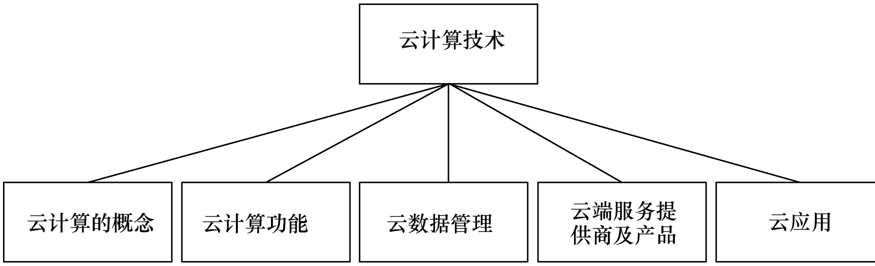


图 1.4 云计算技术

称这种云为语义云。

1.5 云计算实验系统

随着云计算的普及，服务提供商面临着越来越大的挑战。他们必须保持大量的异构数据，同时提供高效的信息检索。因此，云计算解决方案的重点是可扩展性和查询效率。为此，我们设计并开发了各种云查询处理实验系统。这些措施包括语义 Web 数据以及云中的关系数据的查询处理。我们还要研究可以托管的应用，如在云中的本体管理和社交网络。

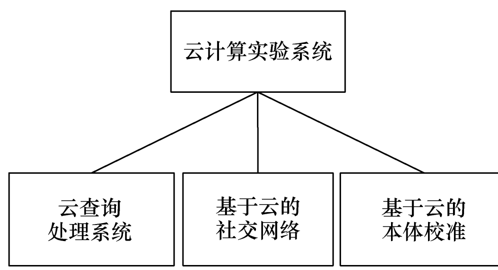


图 1.5 云计算实验系统

本书的第四部分将讨论基于云计算技术开发的实验系统。我们将论述使用 Hadoop/Map/Reduce 框架开发的云查询处理系统。我们的系统处理大量的语义 Web 数据。特别是，我们为 SPARQL 协议和 RDF 查询语言（SPARQL）查询处理器，设计并开发了一个在云端的查询处理器。我们开发了支持社交网络和本体管理应用的云系统。图 1.5 说明了我们开发的一些云计算实验系统。

1.6 云计算安全

目前的云计算技术，如利用 HDFS 和 Map/Reduce 的云计算技术，由于没有提供足够的安全机制来保护敏感数据，所以是不成熟的。因此，我们从空军科学研究室（Air Force Office of Scientific Research，AFOSR）获得资助，并利用最先进的硬件、软件和数据，组成一个安全的云计算平台，用来弥补当前云计算系统开发的不足之处。特别地，我们的目标是①利用现代硬件部件（例如，安全协处理器）来提高性能，这是由于结合了附加的安全功能；②集成开源软件部件以及定制开发软件的部件，来支持复杂数据查询操作；③支持细粒度访问控制和基准监控支持，为复杂数据提供安全性；④为云计

算提供强大的身份验证机制。

我们将在本书的第五部分描述云计算系统的安全性。特别地，我们将详述①云计算安全框架；②云计算功能安全，如虚拟化安全；③云查询处理安全；④云计算安全产品。云计算安全技术如图1.6所示。

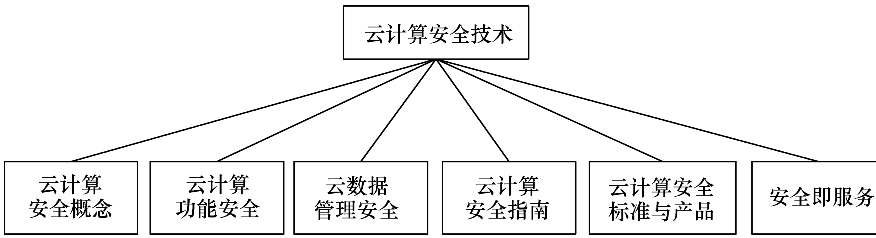


图1.6　云计算安全技术

1.7　云计算安全实验系统

在AFOSR的资助下，我们探索了云计算的安全性。同时，国家科学基金会建立了基础设施，来开展云计算安全的教育课程。基于研究和教育的目的，我们开发了一些云计算安全实验系统。将在本书的第六部分详述我们的一些系统。在图1.7中，我们列出了这些系统。

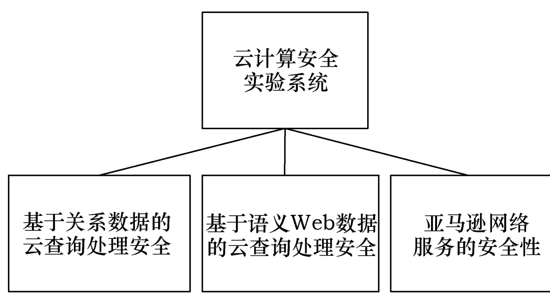


图1.7　云计算安全实验系统

第一个系统对关系型数据系统进行基于云的安全查询处理。第二个系统对语义Web数据进行基于云的安全查询处理。我们基本上利用了上述系统，在本书的第四部分将讨论这些系统，并将安全性纳入这些系统。其次是云计算安全系统，我们开发它时利用了亚马逊安全的云服务。

1.8　面向安全应用的云计算实验

虽然前面几节讨论了云技术、云安全技术以及我们开发的一些实验系统，但在本节中，我们将讨论各种安全应用如何通过使用云受益。其中一些应用将在本书的第七部分进行阐述。

我们的第一个应用是恶意软件检测。我们必须挖掘大量的数据来检测系统是否因恶意软件而遭到破坏。通常情况下，恶意软件可能会改变其模式，以避免被发现。我们已经开发了一个基于云计算的恶意软件检测解决方案。我们开发的第二个应用是基于云的

内部威胁检测。由于恐怖主义和企业之间的激烈竞争，内部威胁成为一个日益危险的问题。我们正在开发用于内部威胁检测的数据挖掘工具。我们将论述一个使用云计算技术的工具。第三个应用是 AIS。各个组织（例如，执法、医疗保健和政府联盟）必须共同努力，共享信息，来解决问题。同时也必须执行适当的安全策略，以便仅共享授权信息。这样的 AIS 应用可以通过使用云而受益匪浅。相关组织机构可以在云中存储数据和策略，并根据策略共享信息。第四个应用是语义云中的 AIS。

其他安全应用可以被托管在云上，包括身份和访问管理以及电子邮件的垃圾邮件过滤。我们在本节中论述的安全应用如图 1.8 所示。请注意，对于这些应用，云提供安全即服务。

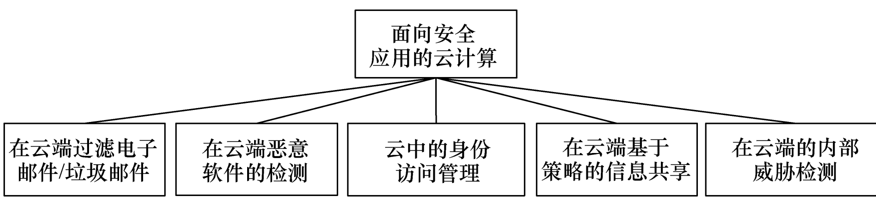


图 1.8 面向安全应用的云计算

1.9 迈向可信赖的云

在我们的定义中，可信赖包括诸如保密性、隐私性、可信任、完整性、高保障、容错性方面，并满足实时约束。本书的重点是云安全，主要介绍的主题是保密性。我们的最终目标是建立可靠的云端。这样的云端必须维护隐私，确保信任，具有高度的完整性，并确保数据质量。请注意，这不是一个标准的定义。也就是说，一些论文和书籍互换使用了可信性和可靠性。此外，一些文件还暗示，安全性包括保密性、完整性和隐私性。

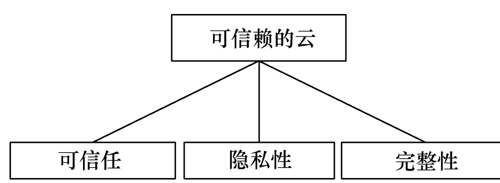


图 1.9 可信赖的云

图 1.9 说明了可信赖的云的各个方面。面临的挑战是确保云提供的所有安全特征，如隐私性、可信任、保密性和完整性。从本质上讲，云必须是灵活的。本书的第八部分将重点关注这样值得信赖的云。

1.10 面向云安全建立基础设施、教育项目和研究计划

虽然在云计算安全方面正在进行技术开发，但同样重要的是在云计算安全上构建安全的基础设施、教育项目和研究计划。图 1.10 说明了我们在这些领域的工作。

我们迫切需要安全地存储、管理、分享和分析大量复杂（例如半结构化和非结构

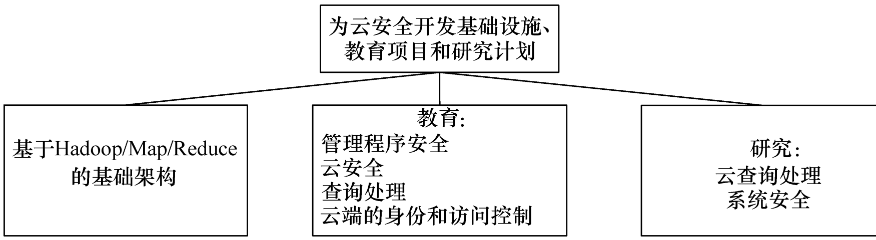


图 1. 10　面向云安全的基础设施、教育项目和研究计划

化）数据，以确定模式和趋势，用来提高医疗保健质量，更好地维护国家和探索可替代的能源。然而，据我们所知，上述所需求的基础设施并没有得到解决。因此，我们在得克萨斯大学达拉斯分校（UTD）开发了一种支持云安全的基础设施，这种设施由 AFOSR 和美国国家科学基金会资助。使用这种基础设施，用户可以查询大量复杂的（例如，语义 Web 和地理空间）数据，同时确保数据的保密性和隐私性。

为了建立有效的云计算安全系统，我们还需要在这个主题上建立一个强大的教育项目，以便我们让学生开展相关工作。因此，我们还在美国国家科学基金会的资助下，通过云计算安全项目设立了包含一些课程的教育项目。这些课程形成了一套复杂的体系，这个体系为其他机构的云计算安全能力建设和教育提供了一个范例。

虽然云计算受到了极大的关注，但在保护云安全方面还有许多工作需要去做。因此，我们启动了由 AFOSR 在 2008 年至 2014 年期间资助的多组织合作研究项目，其中包括对云安全问题的研究，例如云存储安全，云数据和信息管理安全，面向云的信息以及安全模型。

1. 11　本书的结构

本书分为九个部分，每个部分描述与构建和保护云相关技术的某些方面。本书的重点主要是关注云查询处理和周边安全问题，同时还将介绍利用云计算的安全应用，如恶意软件检测、AIS 和内部威胁检测。

第一部分，由 3 章（第 2 ~ 4 章）组成，将描述云的支撑技术。这些包括第 2 章论述的计算系统的演进，第 3 章介绍的安全技术以及第 4 章阐述的数据、信息和知识管理技术。这些支撑技术为云数据管理和云计算安全提供了背景信息。

第二部分，由 3 章（第 5 ~ 7 章）组成，将论述服务和服务安全技术的概念。Web 服务是云计算的核心，因此，理解 Web 服务的安全性问题对于云计算安全很重要。这是因为云为其客户提供服务，包括基础设施服务、平台服务和应用服务。第 5 章介绍 SOA 和 Web 服务、SOAD 以及安全服务。第 6 章将讨论语义 Web 服务和相关的安全问题。第 7 章将论述专用的网络服务，如信息和知识管理服务。

第三部分，由 5 章（第 8 ~ 12 章）组成，将论述云计算的核心课题。在第 8 章中，介绍基本的云计算概念和云计算框架的实现。第 9 章将阐述包括虚拟化、存储管理和数

据管理在内的云计算功能。我们将用整个第 10 章论述云数据管理，因为我们开发的许多原型集中在云数据管理。第 11 章讨论云应用。第 12 章讨论各种云产品和服务提供商。

第四部分，由 3 章（第 13 ~ 15 章）组成，将讨论我们开发的云计算实验系统的原型。第一个是云查询处理系统原型，在第 13 章讨论。第 14 章介绍云中的社交网络。第 15 章讨论我们开发的多个云计算系统，包括基于云的社交网络、基于云的语义 Web 数据管理和基于本体的云查询处理。

第五部分，由 6 章（第 16 ~ 21 章）组成，讨论云计算安全。第 16 章介绍云计算的安全性。云计算功能的安全，如虚拟化安全、存储安全和数据管理安全在第 17 章中讨论。由于我们的大部分工作都集中在云数据管理安全，所以我们将用整个第 18 章讨论这个主题。云计算安全指南将在第 19 章中说明。安全即服务的概念在第 20 章中阐述。最后，云计算产品安全将在第 21 章中讨论。

第六部分，由 3 章（第 22 ~ 24 章）组成，将介绍我们开发的云数据管理安全的实验系统。使用关系数据进行云查询处理的安全是第 22 章的主题。第 23 章将讨论具有语义 Web 数据的查询处理安全。第 24 章介绍亚马逊 Web 服务和信息集成的安全性。

第七部分，由 4 章（第 25 ~ 28 章）组成，将介绍使用云计算的安全应用。第 25 章描述云在恶意软件检测中的使用。基于云的内部威胁检测将在第 26 章讨论。第 27 章阐述了基于云的信息共享保障。第 28 章介绍基于语义云的 AIS。

第二 ~ 七部分讨论的安全问题主要集中在安全性的保密方面，在第八部分，由 3 章（第 29 ~ 31 章）组成，将讨论我们关于建立可信赖的云的一些想法。这种云不仅要确保敏感数据的安全，还必须确保可信任、隐私性和完整性。在第 29 章中，讨论云服务的信任管理。在第 30 章中，介绍云服务的隐私问题。在第 31 章中，阐述云服务的完整性和数据质量。

最后，第九部分，由 4 章（第 32 ~ 35 章）组成，讨论我们关于开发云安全的基础设施、教育项目和研究计划的方法。第 32 章描述一个云安全的基础设施，包括硬件基础设施、软件基础设施和数据基础设施。第 33 章介绍我们正在开发的云计算安全的教育项目。它包括云计算安全的顶级课程，以及对当前安全课程的增强，如数据和应用安全性，以及其相关云计算安全主题单元（例如，云数据管理安全）。在第 34 章中，将描述我们关于云安全的研究合作情况。我们已经为云安全定义了一个框架，并且已经确定了几个研究问题，同时正在为这些问题开发解决方案。第 35 章对本书进行了总结，并讨论了将来的研究方向。

每一部分的内容都是以简介开始，以总结结尾。此外，第 2 ~ 34 章分别以概述开始，并以该章总结和展望以及参考文献结尾。另外，本书还有 4 个附录，用以补充书中讨论的概念。附录 A 提供了数据管理的概述，并讨论我们所写章节之间的关系。这是我们所有书籍的标准做法。在附录 B 中，我们讨论数据挖掘技术。这是因为我们云计算的大部分工作都集中于云数据管理，以及应用数据挖掘技术，来进行云中的恶意软件检测和内部威胁检测。云数据挖掘有时被称为“大数据分析”，不仅可用于恶意软件检测，还可用于医疗保健管理、财务分析和科学数据管理。附录 C 中描述数据安全的一

些重要概念，如访问控制和查询修改（也称为查询重写）。在整本书中，我们将许多这些概念应用于云数据管理安全。由于我们已经在前几章介绍了基于云的 AIS，所以我们在附录 D 中说明一下 AIS 项目。

我们开发了一个 9 层架构来更好地解释本书的概念。这个架构如图 1.11 所示。第 1 层是支撑技术层，涵盖了本书的第一部分。第 2 层是安全服务层，涵盖本书第二部分。第 3 层是云计算层，涵盖本书第三部分。第 4 层是实验系统层，涵盖本书第四部分。第 5 层是云安全技术层，涵盖本书第五部分。第 6 层是云安全实验系统层，涵盖本书第六部分。第 7 层是云应用层，涵盖本书第七部分。第 8 层是可信赖的云层，涵盖本书第八部分。第 9 层是云的基础设施、教育项目和研究层，涵盖本书第九部分。本书的章节和框架的组成部分之间的关系如图 1.12 所示。

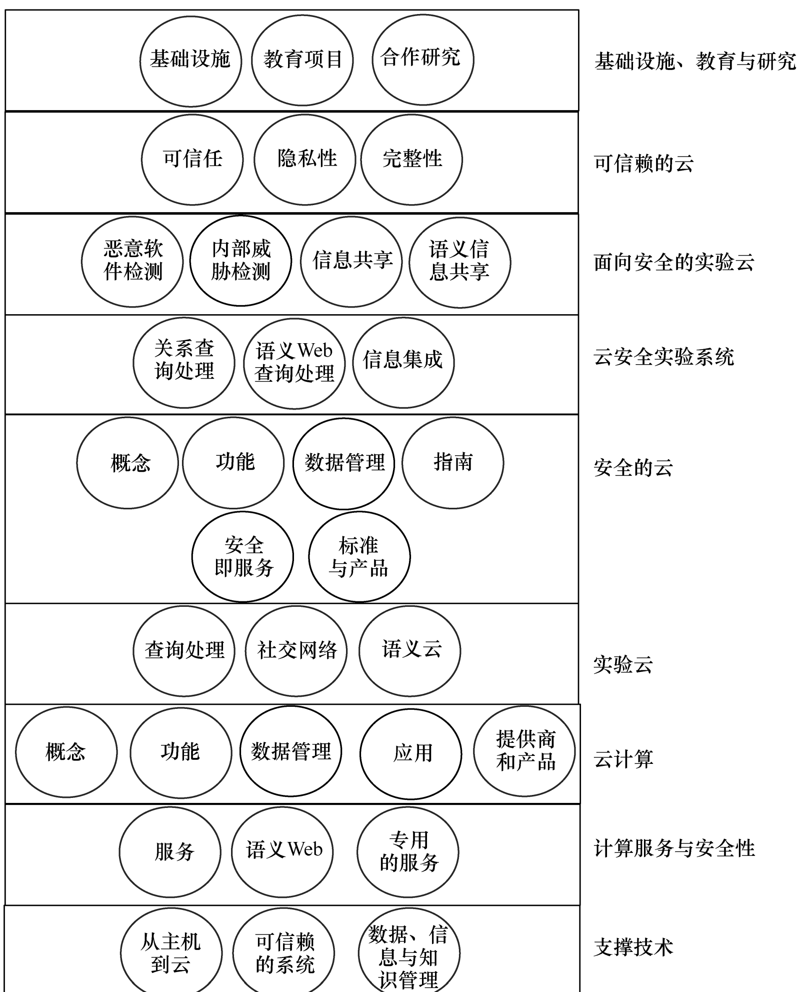


图 1.11　云计算安全分层框架

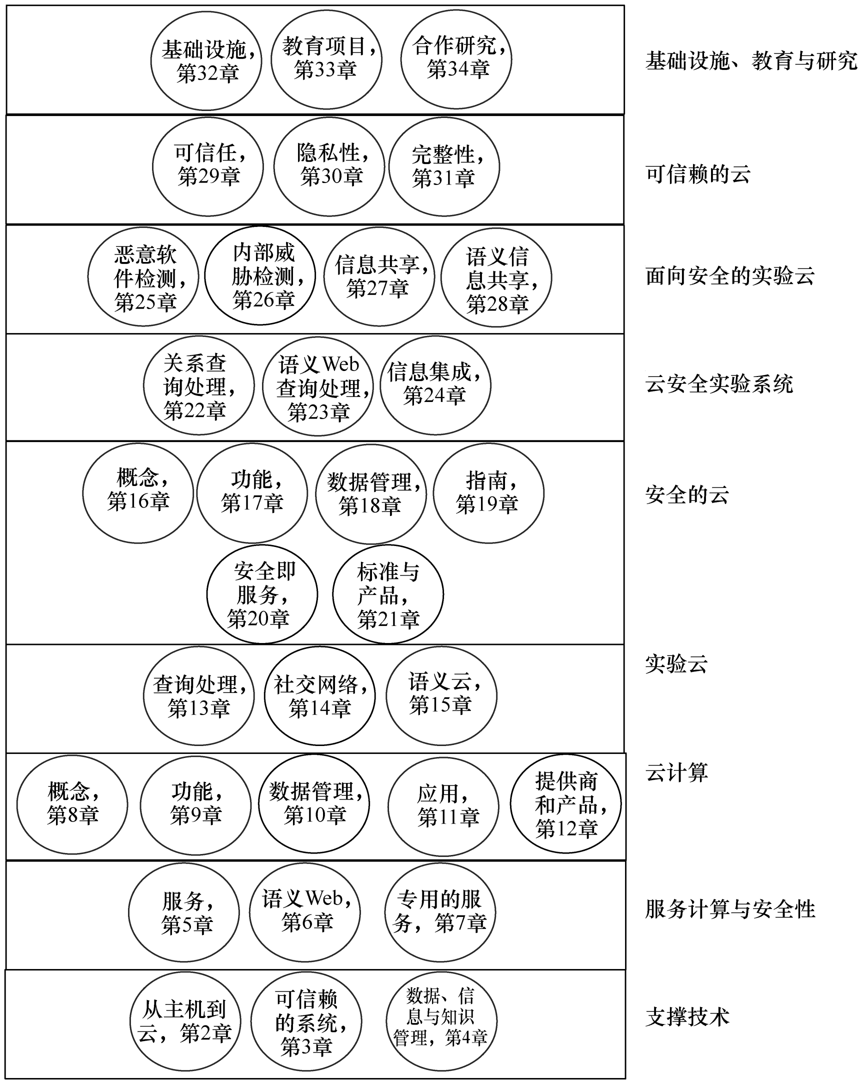


图 1. 12 本书的内容架构

1.12 后续

本章主要介绍本书的构成。首先，简要概述了云安全技术的支撑技术，包括安全性和数据管理。接下来，讨论了网络服务安全和语义 Web 服务安全，其次是云技术和云安全技术的论述，然后是论述利用云的安全应用，接着是关于值得信赖的云的讨论。最后，我们对构建云的安全基础设施、教育项目以及研究合作的方法进行了论述。

本书提供了读者了解云计算和云计算安全的信息。我们对这个话题进行了大量的研究，深入讨论了云查询处理安全等主题。其他一些话题只是进行了简要讨论，如虚拟化安全和云取证。但是，我们为这些主题提供了参考文献。然而，应该指出的是，我们讨论的许多话题还处于研究阶段。

本书的主要贡献之一是提高云计算安全重要性的意识。也就是说，我们与之前所出版的书一致，特别是，为了向技术管理者解释什么是云计算安全。我们在云计算安全方面进行了相当广泛的研究，还试图涵盖包括技术细节以及实验系统的内容，这也会帮助到技术人员和研究人员。我们在本书中也提供了参考资料。

近年来出现了几个新的云计算会议。这些包括与服务计算会议以及 CloudCom 一起举办的 IEEE（Institute of Electrical and Electronic Engineers，美国电气电子工程师学会）计算机学会的云计算会议。计算机协会（Association of Computing Machinery，ACM）最近开始举行自己的云计算会议。我们认为，随着云计算安全技术的进步，将出现完全致力于云计算安全的会议和期刊。我们鼓励读者跟上这个快速增长的发展态势。我们坚信未来的计算是云计算。服务和组件将由来自世界各地的多个提供商开发。而挑战是将这些服务和组件放在一起，并构建安全的云计算系统和应用。还有一个挑战是，即使部分云可能被破坏，但还要确保云运行，并为其客户提供安全的服务。

参考文献

[NIST] The NIST definition of Cloud Computing, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

第一部分　支撑技术

第一部分简介

为了使云计算能安全地应用，我们需要确保多种支撑技术必须协同工作。这些措施包括计算技术，安全技术和数据、信息与知识管理技术。我们将在第一部分讨论这些支撑技术。

第一部分由 3 章组成，即第 2 章、第 3 章、第 4 章。在第 2 章中，我们对计算技术进行概述。具体来说，将论述从大型主机到云的计算演进。在第 3 章中，我们将讨论安全技术。特别是，对设计安全系统方面进行讨论。在第 4 章，我们将讨论数据、信息与知识管理技术。请注意，我们的许多云计算实验系统都是基于数据、信息与知识管理技术的。

第 2 章　从主机到云

2.1　概述

虽然早期的计算系统是基于查尔斯·巴贝奇的机械系统，但现代电子计算机系统在冯·诺依曼的工作下诞生于 20 世纪 40 年代。这是 20 世纪 30 年代著名的数学家和逻辑学家库尔特·哥德尔、艾伦·图灵和阿隆佐·邱奇等进行研究的结果。这些数学家和逻辑学家着重于可计算性的概念。也就是说，能否提前确定程序是否停止？这就是著名的艾伦·图灵的停机问题。

自 20 世纪 40 年代电子计算机的问世，到 20 世纪 50 年代的大型计算机，再到 21 世纪初的云主机，电子计算机已经取得了巨大的进步。本章概述了计算系统的演进。在 2.2 节中，介绍了大型计算机、微型计算机和个人计算机。在 2.3 节中，论述了分布式计算，包括分布式对象管理。在 2.4 节中，讨论了 Web 的出现。在 2.5 节中，讨论了云的出现。2.6 节总结本章。图 2.1 说明了这种演进。

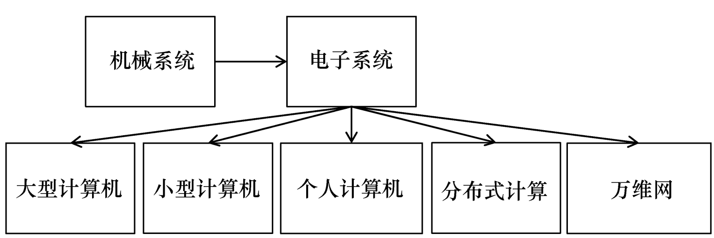


图 2.1　计算系统的演进

2.2　早期计算系统

大型计算机出现在 20 世纪 50 年代，并在 20 世纪 60 年代得到了关注。诸如国际商业机器（International Business Machines，IBM）、Univac、数字设备公司（Digital Equipment Corporation，DEC）和控制数据公司等公司开始开发强大的主机系统。这些主机系统主要为科学家和工程师进行数字运算，使用的主要编程语言是 Fortran。而后在 20 世纪 60 年代，数据库系统的概念产生，一些公司也开发了基于网络和分层数据模型的数据库系统。当时的数据库应用主要是用 COBOL 编写的。

在 20 世纪 70 年代，像 DEC 这样的公司推出了微型计算机的概念。DEC 的 VAX 机器就是一个例子。这些机器比大型主机系统小得多。就在那段时间，终端被开发了出

来。这样，程序员就不必去计算中心，就能使用穿孔卡进行计算。他们可以使用他们的终端，并将作业提交到计算机。这是一个巨大的进步。也正是在这段时间里，C 语言和 UNIX 等操作系统被开发出来。

20 世纪 70 年代末的一个重大发展是个人计算机的出现。苹果电脑因此诞生了。不久之后，IBM 开发了自己的个人计算机。微软为这些 IBM 的机器开发了 DOS。在 20 世纪 80 年代初，一些公司如 Sun 微系统、阿波罗和 HP（惠普）等开发了强大的工作站。基于关系数据模型的数据库系统由 IBM 和 Oracle 等企业开发。到了 80 年代中期，计算机已经做好全面服务于世界的准备。图 2. 2 说明了早期计算系统。

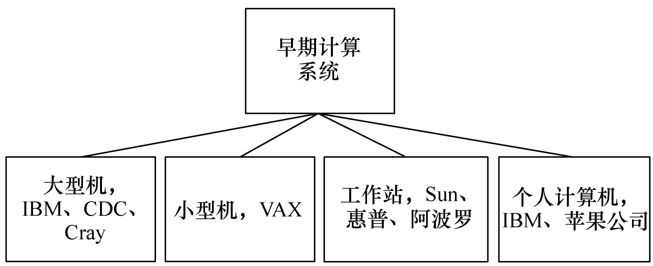


图 2. 2　早期计算系统

2. 3　分布式计算

随着国防高级研究计划局（Defense Advanced Research Projects Agency，DARPA）发明了互联网，网络系统在 20 世纪 70 年代获得了发展机会，早期的产品出现在 20 世纪 80 年代。计算机通过网络联系在一起，并且通过现在称为电子邮件的方式进行通信并交换消息。同时也为这些分布式系统开发了一些应用程序。这个思想就是在多台机器上利用资源并进行计算。20 世纪 80 年代后期，出现了并行计算。

20 世纪 90 年代初期出现的计算模式是分布式对象模式。在这里，计算机被封装为对象。通过这种方式，对象通过交换消息进行通信。这项工作使得诸如对象管理组（Object Management Group，OMG）等联盟成立。正是在这个时候，面向对象语言（例如，Smalltalk 和 C + +）崭露头角。图 2. 3 说明了分布式对象计算。

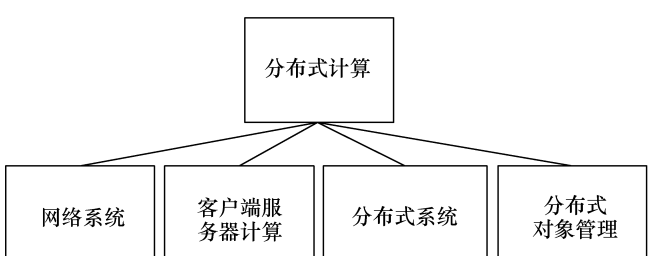


图 2. 3　分布式计算

2.4 万维网

20 世纪 90 年代初，20 世纪的主要创新之一出现了，就是万维网（WWW）。WWW 的发明者蒂姆·伯纳斯·李是 CERN（欧洲核子研究委员会，Conseil Europeen pour la Recherche Nucleaire）在瑞士日内瓦的程序员。他发起了一个支持物理学家共享数据的项目。这个项目促使了万维网的产生。大约在同一时间，伊利诺伊大学国家计算中心的程序员开发了 MOSAIC 浏览器。这两个创新使得普通百姓可以使用万维网查询和搜索信息。20 世纪 90 年代末，出现了几个搜索引擎，如 AltaVista 和 Lycos。然后来自斯坦福大学的两名研究人员创办了一家名为谷歌的公司，该公司目前是世界上著名的网络搜索公司。Java 也成为一个流行的编程语言。

20 世纪 90 年代也预见到了现在的网络繁荣。一些提供服务的公司的成立，促使电子商务的产生。然而，当时的基础设施技术还不成熟，因此，这些公司许多都没有生存下来。在 20 世纪 90 年代末和 21 世纪初，基于服务模式的网络服务概念被创造出来。通过服务技术，为电子商务建立了更好的基础设施。一些公司根据服务模式向消费者提供服务。

20 世纪 90 年代末和 21 世纪初期，语义 Web 技术也出现了。蒂姆·伯纳斯·李的愿景是建立一个更智能的网络，并且这个网络可以理解网页。他创建了标记语言，如 XML 和 RDF；还创建了本体论，以便不同的社区可以相互理解。服务计算和语义 Web 的发展催生了社交媒体。Facebook 等公司成立于 21 世纪 00 年代的中期，使普通人不仅可以在网络上搜索信息，还可以通过网络进行信息交流和信息共享。图 2.4 说明了万维网的演进。

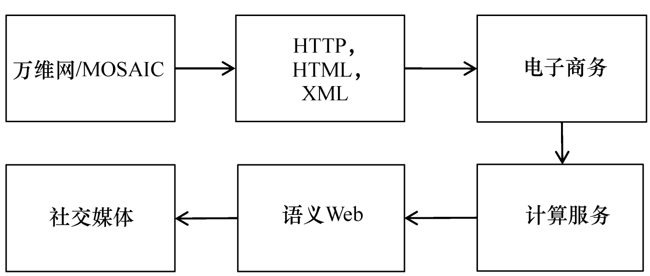


图 2.4　万维网的演进

2.5 云计算

服务计算、分布式计算和万维网的发展促使了云计算的产生。这个想法是提供计算作为服务，就像我们使用电力服务一样。也就是说，云服务提供商将为消费者提供不同程度的服务。该服务可以使用云去计算，去进行数据库管理，或为应用程序提供支持，如组织财务。

本书的第三部分、第四部分、第五部分和第六部分将着重介绍云计算的概念、实验系统和云的安全性。由于服务计算是云计算的一个重要方面，所以我们将用第二部分来介绍服务计算。特别地，我们讨论了网络服务和网络服务的安全性。图 2. 5 说明了云计算的概念。

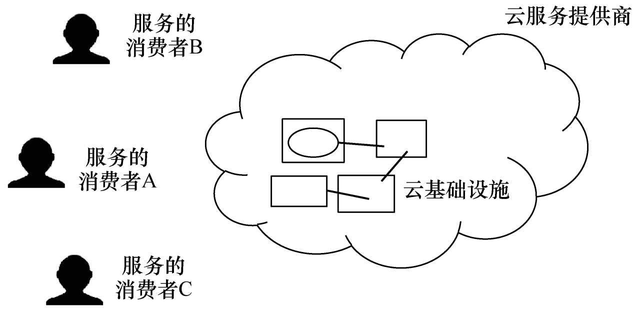


图 2. 5　云计算

2. 6　总结和展望

在本章中，我们简洁地概述了计算的演进。计算机的发展开始于冯 · 诺依曼机器之后数学逻辑学家的研究。然后，我们谈到了计算机从大型主机到个人计算机的演进过程。最后，我们对分布式计算、服务计算、万维网和云计算进行了概述。

本章的论述将为本书中讨论的主题奠定基础。首先，我们将提供一些关于安全系统以及数据、信息和知识管理系统的背景叙述。随后将讨论安全的 Web 服务和可信赖的语义 Web。然后，我们将重点关注云计算和云计算安全的主要议题。特别是，我们将叙述我们为云计算和云计算安全开发的技术、产品以及实验系统。

我们相信，计算会继续前进和发展，同时面临的挑战将是处理大量数据。这一挑战被称为“大数据”问题。因此，利用云计算开发可扩展方案来解决大数据问题，这将成为未来的重点。

参 考 文 献

[BABB] The Babbage engine, http://www.computerhistory.org/babbage/

[HEIJ67] J. Van Heijenoort, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, Harvard University Press, Cambridge, MA, 1967.

[HIST] Computer history, http://www.computerhope.com/history/

[NEUM] von Neumann machine, http://www.britannica.com/EBchecked/topic/1252440/von-Neumann-machine

[OMG] Object Management Group, http://www.omg.org

[TURI] Barker-Plummer, David, Turing machines, http://plato.stanford.edu/entries/turing-machine/

[W3C] World Wide Web Consortium, http://www.w3c.org

第 3 章　可信赖的系统

3.1　概述

正如我们在第 1 章中所述，云计算安全集成了云计算技术和安全技术。我们将在本书的第三部分讨论云计算技术。在本章中，我们将讨论安全技术。特别是，我们将概述可信赖的信息系统，包括安全性、隐私性、完整性、可靠性和实时处理。可信赖的系统是安全可靠的系统。我们指的可靠的系统是具有高完整性、容错性和满足实时约束的系统。换句话说，对于一个可信赖的系统，它必须是安全的、容错的、满足按时调度的，并管理高质量的数据。

本章概述了各种可靠系统的发展，特别强调了包括安全数据系统在内的安全系统。我们专注于安全数据系统，因此我们的几个云实验系统专注于云中的查询处理安全。本章的结构如下：在 3.2 节，我们将讨论安全系统的一些细节。3.3 节提供可靠系统的概述，其中包括信任、隐私、完整性、数据质量、高度保障系统、实时处理和容错能力。在 3.4 节，讨论了一些安全威胁，以及对其正在提出的解决方案。在 3.5 节，讨论了与端到端的安全对应的、用不受信任的组件构建的安全系统。在 3.6 节对本章进行了总结。图 3.1 说明了本章中讨论的概念。

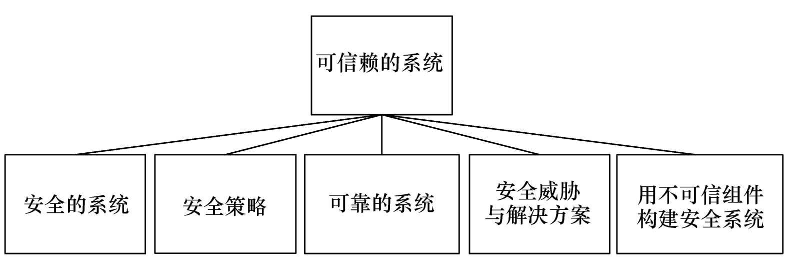


图 3.1　可信赖的系统

3.2　安全系统

3.2.1　简介

安全系统包括安全操作系统、安全数据管理系统、安全网络以及其他类型的系统，如基于 Web 的安全系统和安全数字图书馆等。本节提供了信息安全的发展概况。

在 3. 2. 2 节中，我们讨论信息系统访问控制等基本概念。3. 2. 3 节提供各种安全系统的概述。安全操作系统将在 3. 2. 4 节中讲述。安全数据库系统将在 3. 2. 5 节中论述。安全网络将在 3. 2. 6 节讨论。3. 2. 7 节的主题是新兴趋势。Web 产生的影响将在 3. 2. 8 节给出。3. 2. 9 节将提供构建安全系统的步骤概述。图 3. 2 说明了本节中讨论的各种概念。

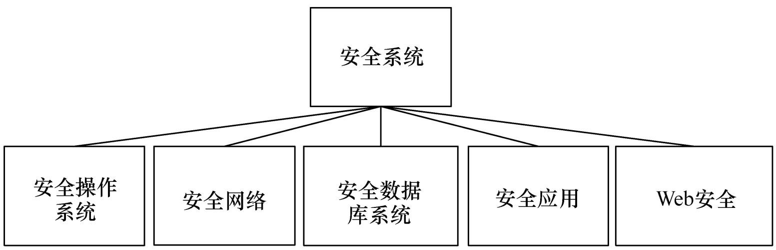


图 3. 2　安全系统

3. 2. 2　访问控制和其他安全概念

访问控制模型包括强制性安全和自主性安全。在本节中，我们将讨论访问控制的两个方面，并考虑其他问题。在自主访问控制模型中，用户或用户组被授予对数据对象的访问权限。这些数据对象可以是文件、关系、对象，甚至是数据项。访问控制策略包括诸如用户 U 具有对关系 R1 的读访问和对关系 R3 的写访问的规则。访问控制也可以包括否定访问控制，如用户 U 没有对关系 R 的读取访问权限。

在强制访问控制中，代表用户行为的主体根据某些策略被授予对对象的访问权限。一个著名的策略是 Bell 和 LaPadula 策略（见本章参考文献［BELL73］），其中主体被授予许可等级，客体具有敏感等级。安全级别组成一个偏序格，为未分类 < 机密 < 秘密 < 绝密。该策略有如下两个属性，即如果主体的许可等级高于客体，则主体可以有读取客体的访问权限；如果客体的许可等级高于主体，那么主体就可以对客体进行写访问权限。还有其他类型的访问控制，如基于角色的访问控制。可以根据用户的角色及其执行的职能，授予用户访问权限。例如，人事经理可以获得工资数据，而项目经理可以访问项目数据。

虽然早期的访问控制策略为操作系统制定，但这些策略已经扩展到包括数据库系统、网络和分布式系统等其他系统。例如，网络策略不仅包括读写的策略，还包括发送和接收消息的策略。其他安全策略包括管理策略。这些策略包括对数据的所有权，以及如何管理和分发数据。数据库管理员以及系统安全人员参与制定管理策略。

安全策略还包括识别和认证策略。每个用户或代理代表用户通过使用一些密码机制进行识别和验证。在分布式系统中，识别和认证变得更为复杂。例如，用户如何在全球范围内进行身份验证？

开发安全系统的步骤包括开发安全策略、开发系统模型、设计系统、审核和确认系

统（见本章参考文献［ANDE01］）。用于审核的方法取决于预期的安全级别。测试和风险分析也是开发过程的一部分。这些活动将确定漏洞以及评估所涉及的风险。图 3.3 说明了各种类型的安全策略。

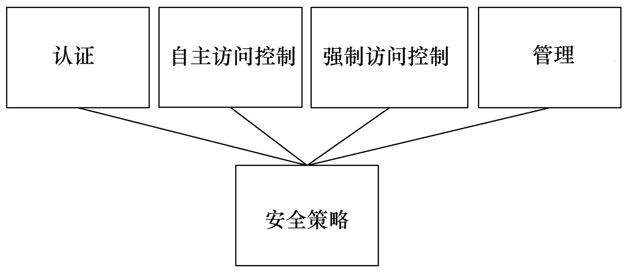


图 3.3　安全策略

3.2.3　安全系统的类型

在 3.2.2 节中，我们讨论了构建安全系统的各种策略。在本节中，将详细介绍各种类型的安全系统。20 世纪 60 年代和 70 年代的大部分早期研究是确保操作系统的安全。为了操作系统的安全，人们制定了早期的安全策略，如 Bell 和 LaPadula 策略。随后，开发出了安全的操作系统，如霍尼韦尔的 SCOMP 和 MULTICS（参见本章参考文献［IEEE83］）。在 20 世纪 80 年代，出现了如基于无干扰模型等其他的策略。

虽然安全数据库系统的早期研究在 20 世纪 70 年代已被报道，但直到 20 世纪 80 年代初这一领域才开始被积极地研究。其中大部分是关注于多层次的安全数据库系统。操作系统的安全策略也略有修改。例如，安全数据库系统的写策略经过修改被声明为，如果主体的级别与客体级别相同，则主体对客体具有写访问。由于数据库系统强制执行数据之间的关系并专注于语义，因此存在额外的安全性问题。例如，可以基于内容、上下文和时间对数据进行分类。进行多个查询，并从合法响应中推断出敏感信息的问题受到关注。这个问题现在被称为推理问题。此外，人们所进行的研究不仅仅在确保关系系统安全上，而且在对象系统以及分布式系统等方面也进行了研究。

计算机网络的研究始于 20 世纪 70 年代末，一直持续到 20 世纪 80 年代甚至更久。网络协议被扩展到纳入安全特性。其结果是安全的网络协议的产生。这些协议包括读、写、发送和接收消息。由于网络和 Web 的出现，加密和密码学的研究引起大量关注。针对独立系统的安全性扩展到分布式系统。这些系统包括分布式数据库和分布式操作系统。许多分布式系统的研究现在集中在保护网络上，称为 Web 安全，也有些研究集中于保护系统，如分布式对象管理系统。

随着新系统的出现，如数据仓库、协同计算系统、多媒体系统和代理系统等，这些系统的安全性将会得到研究。随着万维网的出现，安全不仅受到政府机构的重视，而且也受到商业机构的重视。随着电子商务的出现，更重要的是要保护公司的知识产权。图 3.4 说明了各种类型的安全系统。

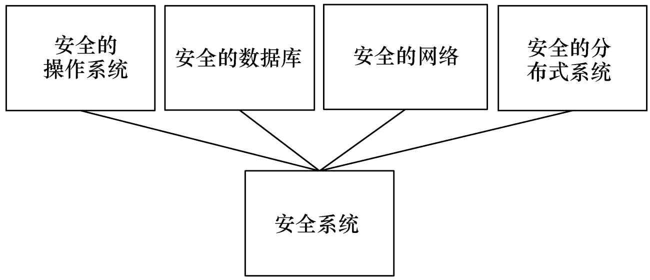


图 3.4　安全系统类型

3.2.4　安全操作系统

操作系统的安全工作在 20 世纪 60 年代和 70 年代广泛开展。随着 Windows、Linux 和其他产品等新型操作系统的出现，研究仍在继续。早期的想法包括访问控制列表和以权限为基础的系统。访问控制列表指定进程的访问类型，其中进程叫主体，文件叫客体。访问通常是读写访问。权限列表是一个进程必须具备访问系统中某些资源的权限。例如，具有特定权限的进程可以写入内存的某些部分。

操作系统强制安全工作开始于 Bell LaPadula 安全模型，它有两个属性：

简单的安全属性规定，如果主体的安全级别高于客体的级别，主体就有对客体的读取访问权。

当主体的安全级别低于客体时，* 属性（称为星属性）规定主体对客体有写入访问权。

从那时起，提出了该模型的演进以及称为无干扰模型的流行模型（参见本章参考文献［GOGU82］）。无干扰模型基本上是更高级别的进程不会干扰较低级别的进程。

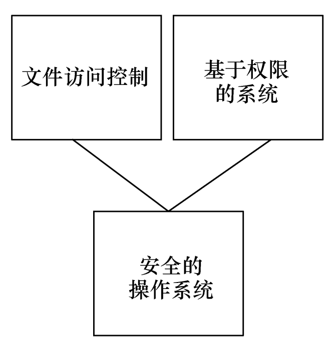


图 3.5　安全操作系统

像微软这样的公司正投入大量资源以确保其产品安全。我们通常会听到各种操作系统的漏洞，以及黑客试图非法进入操作系统，尤其是那些具有网络功能的操作系统。因此，这是未来几年将持续受到关注的一个领域。这是因为包括数据库系统在内的许多应用程序通常托管在操作系统上。因此，如果操作系统被破坏，则可能会影响整个计算系统。图 3.5 说明了操作系统安全性的一些主要方面。

3.2.5　安全数据库系统

数据库的自主安全性工作始于 20 世纪 70 年代，IBM Almaden 研究中心对 R 系统进行了安全性研究。本质上，安全属性指定用户可能对关系、属性和数据元素具有的读取和写入访问。后来，在关系系统（见本章参考文献［GRIF76］）中提出了查询修改等技术，用于访问控制。在 20 世纪 80 年代和 90 年代，对对象系统进行了安全问题的研究。

其中的安全属性用来指定用户对对象、实例变量和类的访问。除了读写访问之外，针对执行访问的方法也进行了说明（见本章参考文献［FERN81］）。

自20世纪80年代初以来，大部分焦点集中在多层次的安全数据库管理系统（见本章参考文献［AFSB83］）。这些系统基本上执行3.2.2节中讨论的强制性策略与在3.3节中描述的修改策略（即可以读取较低或同等级别则，写同等级别的安全策略）。自20世纪80年代以来，已经开发了多层次数据库系统的各种设计、原型和商业产品。Ferrari和Thuraisingham对一些发展进行了详细调查（见本章参考文献［FERR00］）。其中的示例包括由SRI国际提供的SeaView模型或者由霍尼韦尔提供的加锁数据视图（LDV模型）。这些扩展是具有安全属性的关系模型。设计一个用户在不同安全级别看到不同的值的模型是一个挑战。例如，在未分类级别的雇员的工资可能是2万元，而在机密级别的雇员的工资可能是5万元。在标准关系模型中，由于完整性属性，不能表示这些不明确的值。

我们注意到，其他类型的数据库系统的多级安全性也取得了一些重大进展。这些包括对象数据库系统的安全性（见本章参考文献［THUR89］）。在此功能中，安全属性指定读取、写入和方法执行策略。在安全并发控制和恢复方面也进行了大量工作。这里的想法是强制执行安全属性，并满足一致性，而且不会有隐蔽通道。人们还对分布式、异构和联合数据库系统的多层次安全性进行了研究。引起重视的另一个领域是推理问题。有关推理问题的详细内容，请参考本章参考文献［THUR93］。对于安全并发控制，我们参考了Atluri、Bertino、Jajodia等人的众多算法（参见本章参考文献［ATLU97］）。有关分布式和异构数据库安全以及联盟数据库安全的信息，请参见本章参考文献［THUR91］和［THUR94］。

随着数据库系统变得越来越复杂，确保这些系统的安全将变得越来越困难。目前的一些工作重点是保护数据仓库、多媒体数据库和网络数据库（参见例如IFIP数据库安全会议系列会议录）。图3.6说明了各种类型的安全数据库系统。由于我们开发的许多云实验系统开展了数据管理安全工作，所以我们将在附录C中描述数据管理安全的概念和技术。

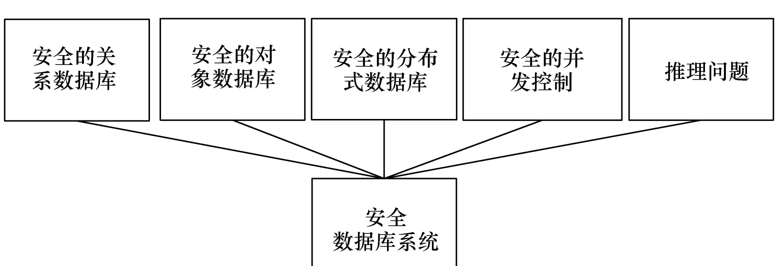


图3.6　安全数据库系统

3.2.6　安全的网络

随着网络的出现和不同的系统与应用之间互连，网络在过去十年中不断扩大。网络

分为公共网络、私有网络、分类网络和未分类的网络。我们经常听到网络被病毒和蠕虫感染的情况。此外，网络也正在被恶意代码和未经授权的人员侵入。因此，网络安全正在成为信息安全的主要领域之一。

我们已经提出了用于网络安全的各种技术。加密和密码应用仍然占主导地位的研究。关于各种加密技术的讨论，大家可以参考本章参考文献［HASS00］。数据挖掘技术正在广泛应用于入侵检测（参见本章参考文献［NING04］）。在网络协议安全性方面也有很多工作，其中安全性被并入到各个层中，诸如网络层、传输层和会话层的协议栈（参见本章参考文献［TANN90］）。核查和验证技术也正在研究用于保护网络。人们还出版了关于这个主题的各种书籍（参见本章参考文献［KAUF02］）。图3.7说明了网络安全技术。

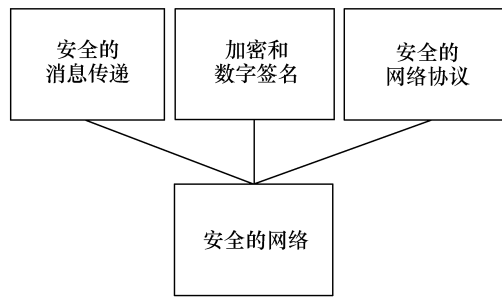


图3.7 安全的网络

3.2.7 新兴趋势

在20世纪90年代中期，系统安全的研究扩展到新兴系统领域。这些包括协作计算系统安全、多媒体计算安全和数据仓库安全。同时，数据挖掘技术引起了新的安全担忧。由于用户现在可以访问各种数据挖掘工具，并且可能会产生敏感关联，这可能会加剧推理问题的严重性。另一方面，数据挖掘也可以帮助诸如入侵检测和审计等安全问题。

网络的到来使得数字图书馆和电子商务安全得到广泛发展。除了开发精细的加密技术，安全研究还侧重于保护Web客户端以及服务器。Java等编程语言的设计考虑了安全性。同时，人们还对安全代理进行了大量研究。

分布式系统安全研究侧重于分布式对象管理系统的安全性。OMG等组织研究了安全属性（见本章参考文献［OMG］）。因此，我们现在拥有商用的安全的分布式对象管理系统。图3.8说明了各种新兴的安全系统和概念。

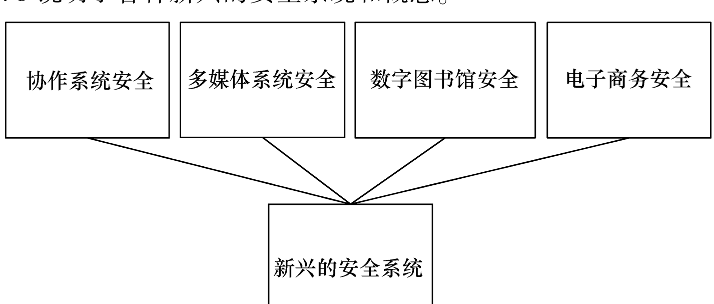


图3.8 新兴的安全系统和概念

3.2.8 网络的影响

网络的出现极大地影响了安全性。现在安全性是主流计算的一部分。政府机构和商业机构担心安全。例如，在金融交易中，如果不保持安全性，则可能会损失数百万美元。随着网络的发展，很容易获取个人的各种信息，因此隐私可能会受到侵犯。

为了保护网络，目前已经提出了许多安全解决方案。除了加密，重点是保护客户端以及服务器的安全。也就是说，必须保证端对端的安全。网络安全也对电子商务有影响。也就是说，当一个人在网络上进行交易时，保证安全是至关重要的。信用卡号码和社会保险号码等信息也必须要得到保护。

上述各节讨论的所有安全问题都必须在网络上考虑。例如，必须制定适当的安全策略。这将是一个挑战，因为没有人拥有整个网络。包括安全的操作系统、安全的数据库系统，安全的网络和安全的分布式系统在内的各种安全系统可以集成在网络环境中。因此，这个集成系统必须是安全的。由于各种数据挖掘工具的推出，推理和隐私等问题可能会加剧。在某些情况下，需要在安全性和其他功能之间进行取舍。也就是说，服务质量是一个重要的考虑因素。除了技术解决方案，法律方面还有待研究。也就是说，律师和工程师必须一起工作。虽然在网络安全方面取得了很大进展，但是随着网络技术的进步，仍还有很多工作要做。图 3.9 显示了网络安全性的各个方面。有关网络安全的讨论，请参阅本章参考文献［GHOS98］。

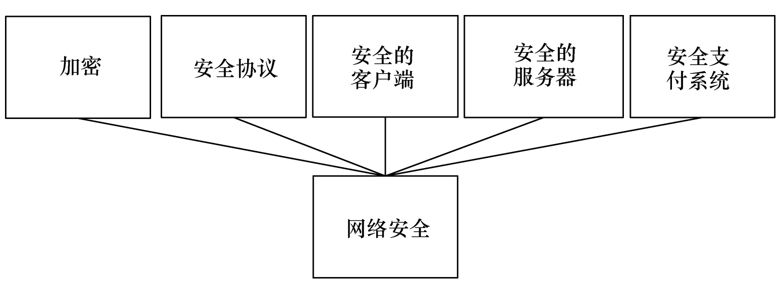


图 3.9　网络安全

3.2.9 构建安全系统的步骤

在本节中，我们将概述构建安全系统的步骤。请注意，我们的讨论是一般性的，适用于任何安全系统。但是，我们可能需要调整各个系统的某些步骤。例如，要构建安全的分布式数据库系统，需要安全的数据库系统以及安全的网络。因此，必须由多个系统组成。

构建安全系统的第一步是制定安全策略。该策略可以用非形式语言表达，然后形式

化。该策略基本上规定了系统必须满足的规则。那么必须开发安全架构，架构将包括安全关键组件。这些是执行安全策略的组件，因此应该被信任。下一步是设计系统。例如，如果系统是数据库系统，则设计查询处理器、事务管理器、存储管理器和元数据管理器模块。同时必须分析系统的设计漏洞。下一阶段是开发阶段。一旦系统实施，它必须进行安全测试。这包括设计测试用例并确保不违反安全策略。此外，根据系统预期的安全级别，可以使用形式验证技术来验证和确认系统。最后，系统将准备好进行评估。请注意，最初的系统要使用可信计算机系统评估标准（见本章参考文献［TCSE85］）进行评估。有关网络标准的解释在本章参考文献［TNI87］，数据库标准的解释在本章参考文献［TDI91］。此外，还可以使用诸如审计和推理控制等各种概念的几个相关文档。请注意，还制定了一些其他标准，包括 20 世纪 90 年代的联邦标准和 21 世纪初的通用标准。

我们还要注意的是，在系统安装于操作环境之前，需要给出环境操作的概念。还必须进行风险评估。一旦安装了系统，就必须对其进行监控，以便检测到包括未经授权的入侵等安全违规。图 3.10 说明了构建步骤。构建安全系统的概述可在本章参考文献［GASS88］中找到。

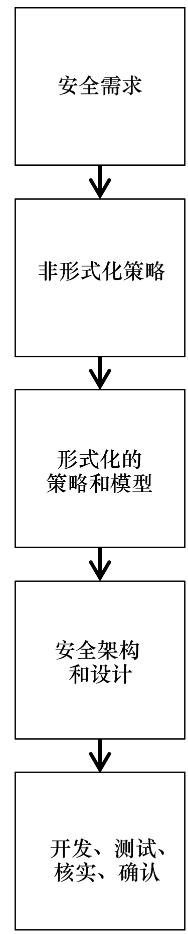


图 3.10　构建安全系统的步骤

3.3　可靠的系统

3.3.1　简介

正如我们在 1.2.2 节讨论的那样，我们的可靠性指的是如信任、隐私、功能的完整性、数据的质量和来源以及权限管理等特征。我们已经分离了保密性，并将其作为安全的一部分。因此，可靠的系统基本上包括安全系统和可靠系统（请注意，这不是标准的定义）。

无论是我们正在讨论的安全性、完整性、隐私、信任，还是讨论的权限管理，总是要涉及成本。那就是说，我们执行安全、隐私和信任的成本是多少？是否可以实施一些

精细的隐私策略和信托管理策略（例如，健康信息便携性和责任法案的隐私策略）？除了让律师和策略制定者与技术人员联系起来，我们还需要引进经济学家的看法。需要进行经济方面的权衡，以实施安全、隐私、信任和权限管理。基本上我们需要的是安全的、保障隐私的、可信任的和拥有权限管理的灵活策略。关于经济对安全的影响的讨论，我们可以参考本章参考文献［NSF03］。

在本节，我们将讨论可靠性的各个方面。3.3.2 节将讨论信任问题。3.3.3 节讨论数字版权管理。3.3.4 节讨论隐私。完整性问题、数据质量和数据来源在3.3.5 节讨论。图 3.11 说明了可靠性的几个方面。

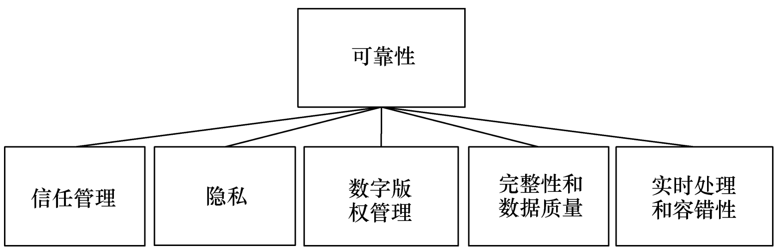


图 3.11　可靠性的几个方面

3.3.2　信任管理

信任管理是关于管理一个个体或群体拥有的对其他人的信任。也就是说，若一个用户有访问数据的权限，我就信任他并给他发送数据吗？用户可能拥有许可或拥有证书，但他可能不值得信赖。信任由用户的行为形成。用户可能违反了信誉或进行了一些不恰当的行为。因此，我可以不信任该用户。现在，即使我不信任约翰，而简信任约翰，她也可以和约翰分享她的资料。也就是说，约翰对吉姆来说可能不值得信赖，但他可能对简来说是值得信任的。

问题是我们如何实施信任？我们可以部分信任某人吗？我们在同一时间可以对约翰有 50% 的信任且对简有 70% 的信任吗？如果我们部分信任某人，那么我们可以分享一些信息吗？我们如何信任我们从比尔收到的数据？也就是说，如果我们不信任比尔，那我们可以相信他给我们的数据吗？信任管理系统以及信任协商系统方面已经有了很多成果。温斯莱特等人开展了广泛的工作，制定了信任的规范语言，以及设计了信任协商系统（见本章参考文献［YU03］）。问题是双方如何协商？如果 B 与 A 共享数据 C，则 A 可以与 B 共享数据 D。如果 B 不与 F 共享此数据，则 A 可以与 B 共享数据 D。有许多这样的规则可以强制执行，这里的困难是开发一个一致执行信任规则或策略的系统。

3.3.3　数字版权管理

与信任管理密切相关的是管理数字版权。这个领域已经被称为数字版权管理（Digital Rights Management，DRM）。这对于娱乐应用尤其重要。谁拥有视频或录音的版权？

权利如何传播？如果侵犯了权利，会发生什么？也就是说，我可以在网络上传播受版权保护的电影和音乐吗？

我们听到很多关于 Napster 和类似组织的争议。DRM 是技术问题还是法律问题？技术专家、律师和策略制定者如何共同合作，才能妥善管理权利？关于 DRM 的大量文章、商讨和争论，本章参考文献［DRM］就是一个有用的信息源。

3.3.4　隐私

隐私是关于保护有关个人的信息。此外，个人可以向 Web 服务提供商指定可以被开放的关于他或她的信息。过去已经对隐私进行了大量的讨论，特别是在保护患者的医疗信息方面。社会科学家和技术专家一直致力于隐私问题。

隐私在近年来受到了极大的关注。这主要是因为网络、语义 Web、反恐和国家安全的需要。例如，为了提取关于各种个人的信息，预防并发现潜在的恐怖袭击，需要检查数据挖掘工具。我们听到很多关于国家安全与媒体的隐私。这主要是因为人们现在意识到要解决恐怖主义的问题，政府可能需要收集有关个人的数据，并挖掘数据来提取信息。数据可能在关系数据库中，也可能在文本、视频和图像中。这正在引起各种公民自由工会的重大关切（参见本章参考文献［THUR03］）。因此，技术人员、决策者、社会科学家和律师正在共同努力，提供解决隐私权被侵犯的方案。

3.3.5　完整性、数据质量和高度保障

完整性是关于维持数据以及相关进程的准确性。数据的准确性被作为数据质量的一部分进行讨论。进程的完整性是确保进程没有被破坏。例如，我们需要确保进程不是恶意进程。由于未经授权的修改，恶意进程可能会损坏数据。为了确保完整性，必须对软件进行测试和验证，以开发高保障系统。

数据库使用者通过实施完整性约束来确保完整性（例如，工资值必须是正数），以及在多个进程访问数据时确保数据的正确性。为了实现正确性，我们执行并发控制等技术。这个想法是强制执行适当的锁程序，以便多个进程不会同时访问数据并损坏数据。

数据质量是确保数据的准确性。数据的准确性可能取决于谁触及了数据。例如，如果数据源不可信，则数据质量值可能较低，实际上一些质量值被分配给每一块的数据。当数据组成时，质量值被分配到数据中，从而使得到的值是原始数据的质量值的函数。

数据来源技术也决定了数据的质量。请注意，数据来源是关于维护历史数据的。这将包括诸如访问数据以进行读/写目的的信息。然后根据这个历史记录，可以为数据分配质量值，并确定数据何时被误用。

其他密切相关的话题包括实时处理和容错。实时处理是关于符合时序约束的过程。例如，如果我们要获得股票报价来购买股票，这时就需要实时获取信息。如果信息到达后但交易平台在当天已经关闭营业，则无济于事。类似地，实时处理技术也必须确保数据是最新的。获得昨天的股票报价不足以做出明智的决定。容错性是确保进程从故障中恢复的。故障可能是发生意外情况或恶意的。在出现故障的情况下，如果进程中止，则

必须重新执行进程的操作。

请注意，为了构建高保障系统，我们需要一个系统来处理故障，需要的系统是安全的并且可处理实时约束。我们在本章参考文献［THUR05a］中已经讨论过，实时处理和安全是互相冲突的目标。例如，恶意进程可以使系统错过关键的时序约束。此外，为了执行所有访问控制检查，一些进程可能会错过最后的调度。因此，我们需要的是灵活的策略，这些策略将决定哪些方面对特定情况至关重要。

3.4 安全威胁和解决方案

近年来，我们听到很多关于病毒和特洛伊木马扰乱网络活动的信息。这些安全威胁和违规行为给企业造成了数百万美元的损失。最近身份窃取的行为也非常猖獗。此外，还发生未经授权的入侵、推理问题和侵犯隐私问题。在本节中，我们将介绍其中的一些威胁。本章参考文献［GHOS98］中也提供了对这些威胁的很好的介绍。在本章，我们将讨论一些正在提出的安全威胁和解决方案。图 3.12 说明了这些威胁。

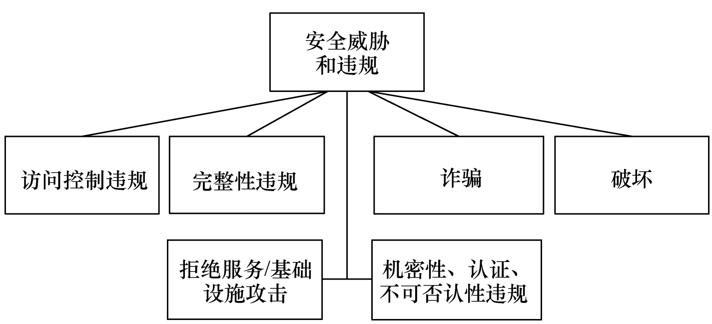


图 3.12　安全威胁

身份验证违规：密码可能被盗，这可能会导致认证违规。人们可能需要有多个密码和关于用户的其他信息来解决这个问题。人们正在采用生物特征和其他技术来处理身份验证违规问题。

不可否认性：一个消息的发送者很可能会否认他已发送的消息。不可否认的技术将确保可以跟踪发送方的消息。如今跟踪消息的所有者并不困难。但是，跟踪访问该网页的人是不容易的。也就是说，尽管已经在分析网络日志上取得了进步，但仍难以确定访问网页的用户的确切位置。

特洛伊木马和病毒：特洛伊木马和病毒是可能导致各种攻击的恶意程序。事实上，本节讨论的许多威胁都可能是由特洛伊木马和病毒造成的。病毒可以从机器传播到机器，并且可以删除各种计算机中的文件。特洛伊木马可能会将信息从较高级别的用户泄露给较低级别的用户。人们已经开发了杀毒软件，并且现在可以购买。

破坏：我们经常听说黑客入侵系统并发布虚假信息。例如，关于各种政府网页被破

坏的一些信息在本章参考文献［GHOS98］中有报道。只需要损坏一个服务器、客户端或网络，这些问题就会波及多台机器。

诈骗：由于大量的商业和贸易在网络上毫无控制地进行，互联网诈骗可能会导致企业损失数百万美元。入侵者可以获得合法用户的身份，并通过伪装可以清空银行账户。

拒绝服务和基础设施攻击：我们听说过黑客正在摧毁基础设施。基础设施可以是电信系统、电力系统或供暖系统。这些系统通常通过网络并由计算机进行控制。这种攻击将导致拒绝服务。

自然灾害：除了恐怖主义，计算机和网络也容易遭受诸如飓风、地震、火灾等自然灾害。数据必须得到保护，数据库必须从灾难中恢复。在某些情况下，自然灾害的解决办法与恐怖袭击造成的威胁相似。例如，使用容错处理技术来恢复数据库。另外，风险分析技术也可能包含损害。

访问控制违规：传统的访问控制违规行为可以扩展到网络。用户可能通过网络访问未经授权的数据。请注意，在网络中，存在许多的数据，控制对这些数据的访问将是一个相当大的挑战。

诚信违规：网络上的数据可能会受到未经授权的修改。此外，数据可能来自任何地方，而且数据的生产者可能不值得信赖。这样可以更容易地破坏数据。不正确的数据可能会导致严重损失，例如不正确的银行账户，这可能导致错误的交易。

保密违规：安全性包括机密性和完整性。也就是说，机密数据必须受到保护，并且不会被清除。需要对数据库的使用进行统计，以防止违规行为发生。

真实性违规：这是一种数据完整性违规的形式。例如，考虑发布商、订阅者和所有者的情况。用户将订阅各种杂志，业主创办杂志（电子版），作为第三方的出版者将发行杂志。如果出版商不值得信任，他可以改变杂志的内容。这违反了文件的真实性。所以，需要通过采取各种解决方案以确定文件的真实性（参见本章参考文献［BERT04］），这些包括加密和数字签名。

侵犯隐私：通过网络可以获得被收集的关于个人的各种信息。另外，利用数据挖掘工具和其他分析工具，可以对个人信息进行各种未经授权的关联分析。

推理问题：推论是冒充查询，并从合法响应中推导未经授权的信息的过程。事实上，我们认为隐私问题是一种推理问题的形式。处理推理问题的许多解决方案已经被提出，包括约束处理和概念结构的使用。我们在3.5节将对其中一些方案进行讨论。

身份盗窃：这几天我们听到很多关于身份被窃取的信息。小偷得到一个社会保障号码，从那里可以窃取个人的银行账户。在这里，小偷变身为合法业主，他现在拥有许多关于业主的关键信息。这是一个非常难以处理和管理的威胁。可行的解决方案尚待开发。数据挖掘有望解决这个问题，但可能技术还不够成熟。

内部问题：内部威胁被认为是相当普遍和相当危险的。在这种情况下，人们永远都不知道恐怖分子是谁。他们可能是数据库管理员或任何被公司信任的人。单独的背景检查可能无法检测内部威胁。对此，我们正在提出基于角色的访问控制以及数据挖掘技术。

这里所有讨论的威胁/攻击加上各种其他网络安全威胁/攻击统统被称为网络恐怖主

义。网络恐怖主义本质上是破坏网络及其所有组件，使敌人或对手的系统崩溃。目前，美国和欧洲各国政府正在投入大量资金，开展保护网络和防止网络恐怖主义的研究。请注意，一般意义上的恐怖主义包括网络恐怖主义、生物恐怖主义和对人身安全的侵害行为，包括轰炸建筑物以及向食物和水源投毒。

为了处理这些威胁，我们提出了各种各样的解决方案，包括防火墙、数据挖掘、加密和访问控制。图 3.13 说明了解决方案。解决方案的细节在本章参考文献［THUR05］、［MASU11］、［THUR93］、［DENN82］和［HASS00］中给出。

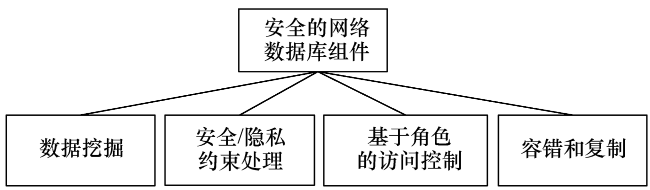


图 3.13　安全性问题解决方案

3.5　通过不可信赖的组件构建安全系统

在 3.2 节中，当我们讨论构建安全系统的步骤时，提出了端到端的安全性。也就是说，我们的论点是要拥有一个真正安全的系统，需要所有的组件（例如操作系统、数据库系统、网络和应用程序）都是安全的。这是因为人们可以设想，安全系统的所有组件将通过单个组织来构建。然而，由于全球化，这一假设已经不再可行。这意味着在组件上不安全的可能性更大。因此，设计师面临的挑战是用不可信的组件构建安全系统。

在最近，我们与其他几所大学一起，对用不可信组件构建安全系统进行了初步研究。我们的挑战是在即使操作系统可能被损坏的情况下，也可设计安全的应用程序。我们的成果在本章参考文献［UTD10］中提供。这项工作也适用于国防部的任务保障。他们的挑战是尽可能长时间地持续执行任务，尽管系统可能会遭到攻击。随着外包的出现，这是未来几年将继续扩大和发展的话题。图 3.14 说明了用不可信的组件来构建安全系统包含的内容。

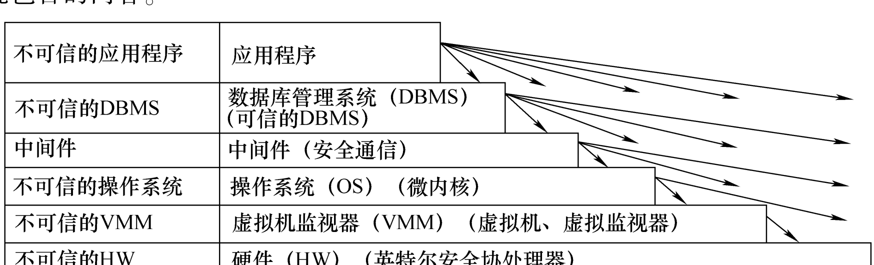


图 3.14　用不可信的组件构建安全系统

3.6 总结和展望

本章简要介绍了可信系统的发展情况。我们首先讨论了安全系统，包括访问控制中的基本概念以及自主和强制性策略，安全系统的类型，如安全操作系统、安全数据库、安全网络和新兴技术，网络的影响以及构建安全系统的步骤。接下来，我们讨论了可靠的系统，包括信任、权限、隐私、完整性、质量和实时处理方面。然后我们关注了安全威胁和解决方案的各个方面的更多细节。最后，我们提供了使用不可信组件来保护系统的概述。

我们虽然在可靠系统方面取得了很大进展，但仍有很多工作要做。我们需要研究新兴系统的安全性，如语义 Web 技术和服务技术。此外，知识管理系统以及地理空间系统的安全性也至关重要。在本书中，我们将重点介绍安全服务方面。如第 1 章所述，云计算安全将安全技术与云计算技术相结合。由于安全的 Web 服务是确保云计算安全的基础，我们将在本书的第二部分讨论安全的 Web 服务。我们将在本书的第三部分和第四部分中描述云计算技术。云计算安全技术和系统将在本书的第五部分和第六部分进行讨论。本书的第七部分讨论云的安全性应用，如恶意软件检测。最后，本书的第八部分将讨论云的信任、隐私和完整性。

参考文献

[AFSB83] *Air Force Studies Board, Committee on Multilevel Data Management Security, Multilevel Data Management Security*, National Academy Press, Washington, DC, 1983.

[ANDE01] Anderson, R., *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, New York, 2001.

[ATLU97] Atluri, V. and S. Jajodia, E. Bertino, Transaction processing in multilevel secure databases with kernelized architectures: Challenges and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 9(5): 693–708, 1997.

[BELL73] Bell, D. and L. LaPadula, *Secure Computer Systems: Mathematical Foundations and Model, M74–244*, The MITRE Corporation, Bedford, MA, 1973.

[BERT04] Bertino, E. et al. Secure selective third party publication of XML data. *IEEE Transactions on Knowledge and Data Engineering*, 16(10): 1263–1278, October 2004.

[DENN82] Denning, D., *Cryptography and Data Security*, Addison-Wesley, MA, 1983.

[DRM] Digital Rights Management Architectures, http://www.dlib.org/dlib/june01/iannella/06iannella.html

[FERN81] Fernandez, E. et al. *Database Security and Integrity*, Addison-Wesley, MA, 1981.

[FERR00] Ferrari, E. and B. Thuraisingham, Secure database systems, In: *Advances in Database Management* (Editors: M. Piatini and O. Diaz), Artech House, UK, 2000.

[GASS88] Gasser, M., *Building a Secure Computer System*, Van Nostrand Reinhold, New York, 1988.

[GHOS98] Ghosh, A., *E-commerce Security, Weak Links and Strong Defenses*, John Wiley, New York, 1998.

[GOGU82] Goguen, J. and J. Meseguer, Security policies and security models, *Proceedings of*

the IEEE Symposium on Security and Privacy, Oakland, CA, April 1983.
[GRIF76] Griffiths, P.P. and B.W. Wade, An authorization mechanism for a relational database system. *ACM Transactions on Database Systems* 1(3): 242–255, 1976.
[HASS00] Hassler, V., *Security Fundamentals for E-Commerce*, Artech House , UK, 2000.
[IEEE83] *IEEE Computer Magazine*, *Special Issue on Computer Security Technology*, 16(7): 26–46, July 1983.
[KAUF02] Kaufmann, C. et al. *Network Security: Private Communication in a Public World*, Prentice Hall, PT, 2002, Upper Saddle River, NJ.
[MASU11] Masud, M., L. Khan, and B. Thuraisingham, *Data Mining for Malware Detection*, CRC Press, Boca Raton, FL, 2011.
[NING04] Ning, P. et al. Techniques and tools for analyzing intrusion alerts. *ACM Transactions on Information and Systems Security*, 7(2): 274–318, 2004.
[NSF03] Proceedings of NSF cyber trust principal investigator meeting, Baltimore, MD, August 2003.
[OMG] The Object Management Group, http://www.omg.org.
[TANN90] Tannenbaum, A., *Computer Networks*, Prentice-Hall, NJ, 1990.
[TCSE85] *Trusted Computer Systems Evaluation Criteria*, National Computer Security Center, MD, 1983.
[TDI91] *Trusted Database Interpretation*, National Computer Security Center, MD, 1991.
[THUR89] Thuraisingham, B., Mandatory security for object-oriented database systems, *Proceedings of the ACM OOPOSLA Conference*, New Orleans, LA, October 1989.
[THUR91] Thuraisingham, B., Multilevel security for distributed database systems. *Computers and Security*, 10, 727–747, 1991.
[THUR93] Thuraisingham, B., W. Ford, and M. Collins, Design and implementation of a database inference controller, *Data and Knowledge Engineering Journal*, 11(3): 271–297, 1993.
[THUR94] Thuraisingham, B., Multilevel security for federated database systems. *Computers and Security*, 13, 509–525, 1994.
[THUR98] Thuraisingham, B., *Data Mining: Technologies, Techniques, Tools and Trends*, CRC Press, Boca Raton, FL, December 1998.
[THUR03] Thuraisingham, B., *Web Data Mining Technologies and Their Applications in Business Intelligence and Counter-Terrorism*, CRC Press, Boca Raton, FL, 2003.
[THUR05a] Thuraisingham, B., *Database and Applications Security: Integrating Data Management and Information Security*, CRC Press, Boca Raton, FL, 2003.
[THUR05b] Thuraisingham, B, Privacy constraint processing in a privacy enhanced database management systems. *Data and Knowledge Engineering Journal*, November, 2003.
[TNI87] *Trusted Network Interpretation*, National Computer Security Center, MD, 1983.
[UTD10] Securing the Execution Environment Applications and Data from Multi-Trusted Components, University of Texas at Dallas, Technical Report, UTDCS-03-10, 2010.
[YU03] Yu, T. and M. Winslett, A unified scheme for resource protection in automated trust negotiation, *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003.

第 4 章　数据、信息和知识管理

4.1　概述

第 2 章讨论了计算系统的演进，第 3 章讨论了可靠系统，本章将讨论数据、信息和知识管理中的一些基本概念。这是因为我们对云实验系统工作的大量讨论，是基于数据、信息和知识管理的安全的。

人们给出了数据、信息和知识的多种定义。我们采用的定义如下：数据是指由数据库管理的数据，信息是从数据库中的数据中提取的数据，知识是被理解的信息。因此，一个人需获得知识才能进行活动。例如，AA126 是数据。当我们说美国航空航班 126，我们从数据中提取了信息。当我们说 AA126 在上午 9 点从波士顿到达拉斯，我们知道如何处理这样的信息，比如预订机票。

本章的组织结构如下：在 4.2 节中，我们将讨论数据管理；信息和知识管理将分别在 4.3 节和 4.4 节中讨论；活动管理如电子商务和信息共享将在 4.5 节讨论。4.6 节进行本章总结。应该指出的是，本章讨论的概念是新兴主题大数据的基础。因此，我们也在 4.6 节对大数据进行了简单的讨论。图 4.1 说明了本章讨论的各种概念。

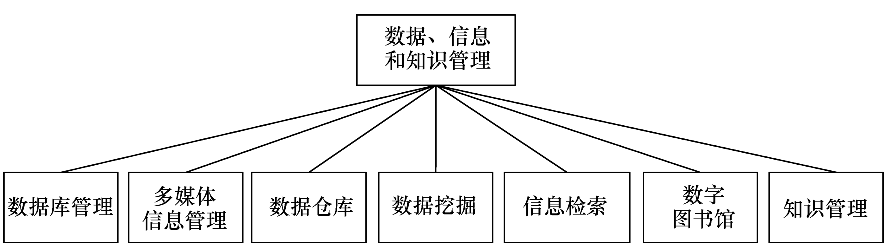


图 4.1　数据、信息、知识和活动管理

4.2　数据管理

在 4.2.1 节中，我们将讨论数据库管理，包括数据模型和数据库功能，另外还有分布式数据管理和 Web 数据管理的讨论。在 4.2.2 节中，我们将讨论复杂的数据管理，包括对多媒体和地理空间数据管理的讨论。

4.2.1　数据管理

4.2.1.1　数据模型

建立数据模型的目的是为了尽可能准确地、完全地、自然地表达它所捕获的世界。

数据模型包括层次模型、网络模型，关系模型、实体关系模型、对象模型和基于逻辑的模型。关系数据模型是数据库系统最流行的数据模型。使用关系模型（见本章参考文献［CODD70］）的数据库被视为关系的集合。每个关系表中都有属性和行。我们已经提出了用各种语言来操纵关系。这些语言中值得注意的是 ANSI 标准 SQL（结构化查询语言）。该语言用于访问和操作关系数据库中的数据（见本章参考文献［SQL3］）。关系数据模型的详细讨论在本章参考文献［DATE90］和［ULLM88］中给出。

4.2.1.2　功能

数据库管理系统（Data Base Management System，DBMS）的功能是执行其操作。DBMS 实际上管理数据库，并支持用户能够查询和更新数据库。因此，DBMS 的基本功能是查询处理和更新处理。在某些应用程序中，例如银行所使用的应用程序中，查询和更新作为交易的一部分。因此，事务管理也是 DBMS 的另一个功能。为了执行这些功能，必须维护数据库中的数据的信息。该信息称为元数据。与管理元数据相关的功能是元数据管理。需要特殊技术来管理实际存储数据的存储空间。与管理这些技术相关的功能是存储管理。为了确保上述功能正确执行，并使用户获取准确的数据，还有一些额外功能。这包括安全管理、完整性管理和故障管理（即容错）。DBMS 的功能架构如图 4.2 所示（另见本章参考文献［ULLM88］）。

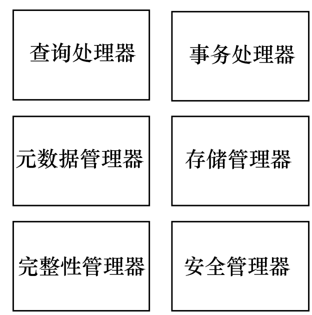


图 4.2　DBMS 架构

4.2.1.3　数据分布

如本章参考文献［CERI84］所述，分布式数据库系统包括分布式数据库管理系统（DDBMS）、分布式数据库和互联网络。DDBMS 管理分布式数据库。分布式数据库是分布在多个数据库中的数据。节点通过通信子系统连接，本地应用程序由本地 DBMS 处理。此外，每个节点还涉及至少一个全局应用程序，因此在此架构中没有集中控制。DBMS 通过称为分布式处理器的组件连接。分布式数据库系统功能包括分布式查询处理、分布式事务管理、分布式元数据管理，以及跨多个节点实施安全性和完整性。语义 Web 可以被认为是大型分布式数据库。

4.2.1.4　Web 数据管理

Web 数据管理研究人员和从业者面临的一个主要挑战是提出一个适当的数据表示方案。问题是：是否需要 Web 数据库系统的标准数据模型？是否有可能制定这样的标准？如果是，标准模型和网络上数据库使用的各个模型之间的关系是什么？

Web 数据库管理功能包括查询处理、元数据管理、安全性和完整性等功能。在本章参考文献［THUR00］中，我们研究了各种 DBMS 功能，并讨论了 Web 数据库访问对这些功能的影响。这里讨论了其中的一些问题。图 4.3 为 Web 数据库功能。查询和浏览是两个关键功能。首先，需要一个适当的查询语言。由于 SQL 是流行的语言，因此可能需要适当的扩展 SQL。从 XML 和 SQL 演变而来的 XML－QL 和 XQuery 正朝着这个方向发展。查询处理涉及开发一个成本模型。互联网数据库管理有特殊的成本模型吗？

关于浏览操作，查询处理技术必须与以下关联的技术相结合。也就是说，超媒体技术必须与数据库管理技术相结合。

事务管理对许多应用程序至关重要。网上可能有新的交易类型。例如，可以通过因特网出售各种物品。在这种情况下，当潜在买家出价时，该物品不应立即被锁定。它在收到几个出价，并且在被出售前必须保持开放状态。也就是说，需要特殊的交易模型。必须为这种交易模型开发适当的并发控制和恢复技术。

元数据管理是 Web 数据管理的主要关注点。问题是：什么是元数据？元数据描述了与库相关的所有信息。这可能包括各种网站、用户类型、访问控制问题和强制执行的策略。元数据应位于哪里？每个参与网站应该保留自己的元数据吗？元数据应该复制还是应该有一个集中的元数据存储库？这种环境中的元数据可能是动态的，这是因为用户和网站可能会不断变化。

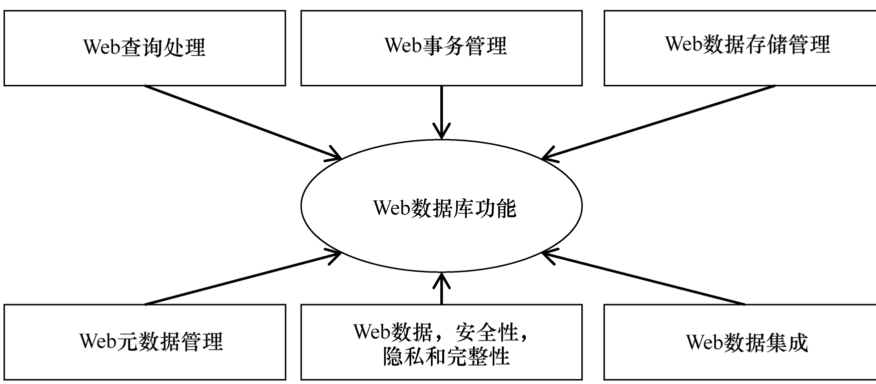


图 4.3　Web 数据库功能

Web 数据库访问的存储管理是一个复杂的功能。需要适当的索引策略和访问方法来处理 Web 数据。另外，由于数据量庞大，还需要将数据库管理技术与大容量存储技术相结合的技术。其他数据管理功能包括集成异构数据库，管理多媒体数据和挖掘。我们在本章参考文献［THUR02a］中讨论它们。

4.2.2　复杂数据管理

4.2.2.1　多媒体数据系统

多媒体数据管理器（MultiMedia Data Manager，MM－DM）提供对来自多媒体数据库的多媒体数据的存储、操作和检索的支持。在某种意义上，多媒体数据库系统是一种异构数据库系统，因为它管理异构数据类型。异构性缘于数据，如文本、视频和音频等多种媒体。由于多媒体数据也传达了演讲、音乐和视频等信息，因此我们将信息分组归档。多媒体数据管理的一个重要方面是数据表示。目前，扩展的关系模型和对象模型已经被提出。

MM－DM 必须为典型的 DBMS 功能提供支持。这些包括查询处理、更新处理、事务管理、存储管理、元数据管理、安全性和完整性。另外，在许多情况下，诸如语音和

视频的各种类型的数据必须被同步显示，因此，实时处理也是 MM－DM 中的主要问题。

为了设计和开发一个 MM－DM，我们正在研究各种架构。在一种架构中，数据管理器用来管理元数据，多媒体文件管理器用于管理多媒体数据。这是一个集成的数据管理和多媒体文件管理模块。在这种情况下，MM－DM 由 3 个模块组成：管理元数据的数据管理器、多媒体文件管理器和用于集成两者的模块。第二种架构是紧耦合方式。在这种架构中，数据管理器管理多媒体数据以及元数据。紧耦合架构具有优势，因为所有数据管理功能都可以应用于多媒体数据库。这包括查询处理、事务管理、元数据管理、存储管理以及安全性和完整性管理。请注意，若使用松散耦合方法，除非文件管理器执行 DBMS 功能，否则 DBMS 仅管理多媒体数据的元数据。

本章参考文献［THUR97］中讨论了其他方面的架构。例如，多媒体数据库系统可以使用诸如面向对象数据库系统的商业数据库系统来管理多媒体对象。然而，对象之间的关系和临时关系的表示可能涉及 DBMS 的扩展。也就是说，DBMS 与扩展层一起提供完整的支持来管理多媒体数据。在另一种情况下，扩展和数据库管理功能都是集成的，因此 DBMS 要管理多媒体对象以及对象之间的关系。本章参考文献［THUR01］讨论了这些架构的进一步细节以及如何管理多媒体数据库。图 4.4 为多媒体信息管理系统。

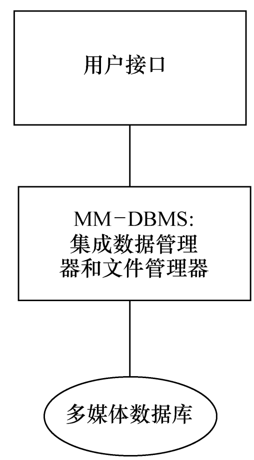


图 4.4　多媒体信息管理系统

4.2.2.2　地理空间数据管理

地理空间数据管理器，通常也被称为地理信息系统（Geographical Information System，GIS），是捕获、存储、分析、管理和呈现与位置相关联的数据的任意系统。如本章参考文献［GIS］所述，地理信息系统是包括具有应用到遥感、土地测量、航空摄影、数学、摄影测量和地理地图软件的系统。GIS 可以被认为是制图和数据库技术的整合。因此，其中的困难是表示空间数据（例如，地图）及存储和查询这些数据。

地理空间数据管理主要得益于开放地理空间联盟（Open Geospatial Consortium，OGC）的活动。除了为地理空间数据开发 XML 的地理空间标记语言（Geospatial Markup Language，GML）之外，OGC 还涉及制定表示、存储和管理地理空间数据的标准。我们为多媒体数据系统（管理文本、语音、视频和音频数据的组合）所描述的许多要求均适用于地理空间系统。其他复杂数据包括传感器数据，用于表示传感器数据的技术，诸如 SensorML 等，也正在被开发。

4.3　信息管理

我们将数据仓库和数据挖掘作为信息管理的一部分，因为这些系统从可能存储在数据库中的原始数据中提取一些有用信息。我们还在信息管理下讨论信息检索和数字图书馆。

4.3.1　数据仓库和数据挖掘

数据仓库是支持数据挖掘和数据分析的关键数据管理技术之一。如由英曼（见本章参考文献［INMO93］）所述，数据仓库是面向对象的。它们的设计在很大程度上取决于利用它们的应用程序。它们集成了多种可能的异构数据源。同时，它们也是持久性的。也就是说，仓库非常像数据库，它们随时间而变化。这是因为随着构建仓库的数据源的更新，在仓库中必须体现这种更改。本质上，数据仓库为企业或组织的决策提供支持功能。例如，虽然数据源可能具有原始数据，但数据仓库可能具有应用原始数据的相关数据、汇总报告和聚合函数。

图4.5为一个数据仓库。数据源由数据库系统A、B和C管理。这些数据库中的信息被合并放入数据仓库。通过数据仓库，数据可能经常被不同的应用程序查看。也就是说，数据是多维的。例如，工资部门可能希望数据处于某种格式，而项目部门可能希望数据处于不同的格式。数据仓库必须支持这种多维数据。

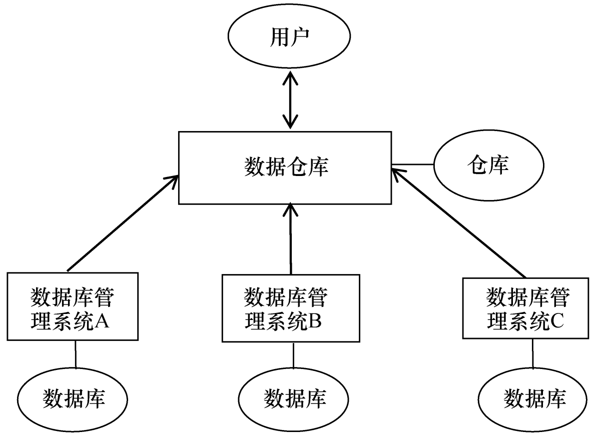


图4.5　数据仓库

数据挖掘是进行多种查询并提取有用的信息、模式、趋势的过程，往往在事先并不知道大量的数据是否在数据库中。实际上，对于许多组织来说，数据挖掘的目标包括：根据过去的经验和当前的趋势来提高营销能力、检测异常模式和预测未来。

一些数据挖掘技术包括基于统计推理、归纳逻辑编程、机器学习、模糊集、神经网络等技术。数据挖掘结果包括分类（查找将数据分成组的规则）、关联（数据之间的关联规则）和排序（查找订单数据的规则）。实际上，人们得到的一些假设，往往是从示例和模式观察中提取的信息。我们需要执行一系列查询，才能观察到这些模式，每个查询可能取决于对先前提出的查询获得的响应。在数据挖掘方面已经有许多进展。在本章参考文献［KDN］中给出了各种工具的讨论，参考文献［BERR97］给出了对结果和技术的详细讨论。图4.6为数据挖掘步骤。我们已经在许多云应用使用了数据挖掘技术，例如恶意软件检测和内部威胁检测，我们将在附录B中提供数据挖掘的概述。

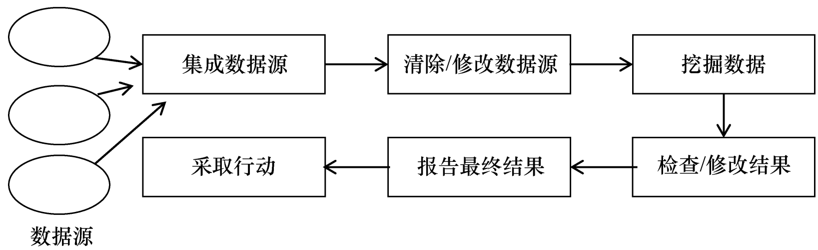


图 4.6　数据挖掘步骤

4.3.2　信息检索

信息检索系统本质上为管理文件提供支持。功能包括文档检索、文档更新和文档存储管理等。这些系统本质上是用于管理文档的 DBMS。这里有许多类型的信息检索系统，它们包括文本检索系统、图像检索系统以及音视频检索系统。图 4.7 为一个通用的信息检索系统，可用于文本检索、图像检索、音频检索和视频检索。这种架构也可以被用于多媒体数据管理系统。每种类型的信息检索系统的特点均在本章参考文献［THUR01］中有所讨论。

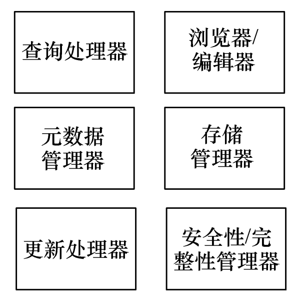


图 4.7　信息检索系统

信息检索系统包括文本检索、图像检索、视频检索和音频检索（见本章参考文献［IEEE03］）。例如，文本检索系统本质上是用于处理文本数据的数据库管理系统。文本数据可以是书籍、期刊、杂志等文件。现在，人们需要一个很好的文档表示数据模型。人们对开发用于文档管理的语义数据模型和对象模型已经进行了大量的工作。例如，一个文件可以有若干个段落，一个段落可以有几个部分等。查询文档可以基于很多因素。我们可以指定关键字，并请求根据关键字检索文档。还可以检索与彼此有某些关系的文档。最近关于信息检索的研究重点是基于语义来查询文档。例如，“检索描述景区视图的文档”或“检索对十岁以下儿童有用的文档”就是这种查询的类型。

信息检索系统的例子是数字图书馆。数字图书馆在国家科学基金会（National Science Foundation，NSF）、国防高级研究计划署（Defense Advanced Research Projects Agency，DARPA）和国家航空航天局（National Aeronautical and Space Administration，NASA）研究的初步努力下，成果表现突出。国家科学基金会继续资助这一领域的特别项目，使得其迅速发展。数字图书馆的理念是数字化所有类型的文件，并提供对这些数字化文件的有效访问。

要使数字图书馆成为现实，必须共同努力发展几种技术。这些包括网络数据管理、标记语言、搜索引擎和问答系统。另外，多媒体信息管理以及信息检索系统也起着重要的作用。本节将回顾一些数字图书馆技术的各种发展。图 4.8 为数字图书馆系统的一个例子。

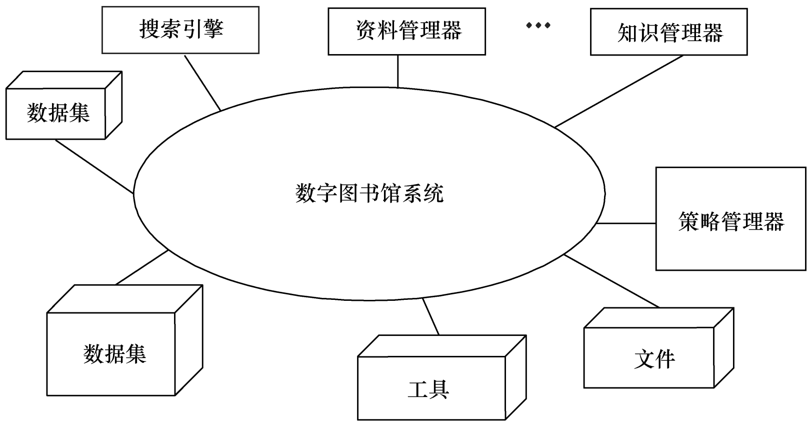


图 4.8 数字图书馆系统

4.3.3 搜索引擎

自 20 世纪 90 年代初以来，已经开发了许多搜索引擎。它们起源于 20 世纪 60 年代及以后开发的信息检索系统。通常当我们调用浏览器时，如 Netscape 或微软的 IE 浏览器时，我们可以访问多个搜索引擎。一些早期的搜索引擎有 AltaVista、雅虎、Info seek 和 Lycos。这些系统诞生于 1995 年前后，对那个时代来说是相当有用的。它们现在有了很大的改善。自 1999 年以来，有一个受欢迎的搜索引擎是 Google。它开始于斯图加特大学研究项目，由 NSF 和美国中央情报局以及行业等组织资助，后来被商业化。Google 以及其他一些搜索引擎都提供智能搜索。然而，在用户可以获得他们的查询的确切答案之前，他们还有很长的路要走。

搜索引擎需要通过浏览器才能访问。当你单击搜索引擎时，你将收到一个请求你输入搜索内容的窗口。然后你要列出正要查找的信息的关键字。然后，搜索引擎将列出各个相关网页的链接。其中的问题是搜索引擎如何找到网页？它基本上在网络上使用信息检索。

搜索引擎的评级由其产生结果的速度决定，更重要的是它产生结果的准确性。也就是说，搜索引擎是否列出了查询的相关网页？例如，当您输入称为“肺癌”的查询时，它是否提供您正在寻找的有关肺癌的相关信息？例如，它可以列出有关肺癌的资源，或者列出关于谁患有肺癌的资料。通常人们想要获得有关肺癌的资源。如果他们想知道谁患有肺癌，那么他们可能会输入“肺癌患者”。

许多搜索的问题虽然非常有用，但却经常提供很多无关紧要的信息。要获得准确的结果，他们必须建立复杂的索引技术。他们还可以缓存来自 Web 服务器的信息，以便应对经常提出的查询。搜索引擎有一个关于他们必须搜索的各种 Web 服务器的目录。当新服务器进入时，此目录将更新。然后搜索引擎构建各种关键字的索引。当用户提出查询时，搜索引擎将参考其知识库，其中包含有关 Web 服务器和各种索引的信息；它

还会检查缓存，如果它没有相关信息，然后才搜索 Web 服务器的信息。所有这一切都必须实时进行。

Web 挖掘使得我们能够挖掘用户日志，并为各种用户构建行为轮廓，从而使搜索更有效率。请注意，现在有成千上万的用户，构建其行为轮廓并不是简单直接的。我们需要挖掘 Web 日志，并找出用户的偏好。然后我们列出用户的这些网页。此外，如果用户正在搜索某些信息，则搜索引擎可能不时地列出可能与用户请求相关的网页。也就是说，搜索引擎必须根据用户想要的内容动态地进行搜索。

4.4 知识管理

知识管理是将知识作为资源来组织管理的过程。这可能意味着分享专业知识、开发一个学习型组织、培养员工、学习经验以及互相合作。知识管理本质上包括数据管理和信息管理。但是，这并不是人人都同意的观点。现在已经提出了各种知识管理的定义。知识管理是一个主要由商学院发明的学科。这些概念已经存在了很长时间了。但是，信息技术（Information Technology，IT）和网络发展的结果造就了知识管理这个词。

在 Morey 等人收集的知识管理论文集（见本章参考文献［MORE01］）中，知识管理分为 3 个领域。也就是建立知识型公司、培养知识型员工、知识管理流程（如技术），包括开发共享文档和工具的方法，以及衡量知识管理效率的指标。在“哈佛商业评论”中有关知识管理的一系列文章，描述了一个知识渊博的公司，建立一个学习型组织，并教授人们如何学习［HARV96］。组织行为和团队动态在知识管理中发挥着重要作用。

知识管理技术包括几种信息管理技术，而信息管理技术包括知识表示和知识库管理系统。其他知识管理技术包括协作工具、用于组织网络信息的工具，以及用于衡量所获得的知识的有效性的工具，如收集各种指标。知识管理技术实际上包括数据管理和信息管理技术以及决策支撑技术。图 4.9 为一些知识管理组件和技术，它还列出了知识管理周期的各个方面。网络技术在知识管理中发挥着重要作用，所以知识管理与网络密切相关。尽管知识管理实践已经存在多年，但确实是网络促进了知识管理。

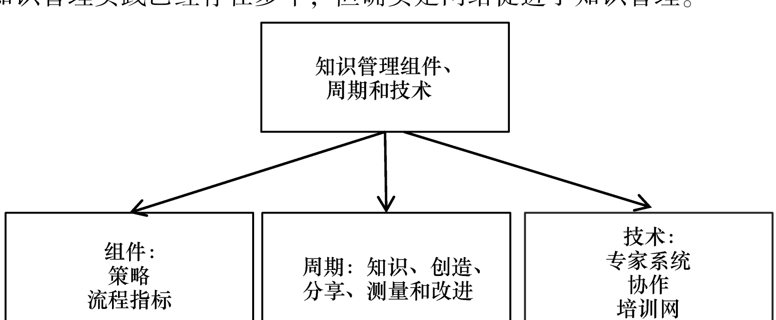


图 4.9　知识管理组件和技术

许多公司现在都有内部网，这是一个最强大的知识管理工具。成千上万的员工通过组织中的网络进行连接。大型企业在世界各地都有网站，员工之间的关系也日益密切。电子邮件可以被认为是早期的知识管理工具之一。而现在有很多工具，如搜索引擎和电子商务工具。

随着网络数据管理和电子商务工具的普及，知识管理将成为网络和电子商务的重要组成部分。本章参考文献［MORE01］中提供了有关知识管理经验的文章，包括策略、流程和指标。协作知识管理在本章参考文献［THUR02b］中介绍。

4.5　活动管理

这里的活动包括电子商务、信息集成、信息共享和供应链管理。我们在本节将讨论这些活动。在第二部分中，我们将介绍如何调用 Web 服务来执行这些活动。在第三部分和第四部分中，我们将讨论如何调用云服务来执行这些活动。

4.5.1　电子商务和电子贸易

我们正在开发各种模型、架构和技术。B2B 的电子商务是关于两家企业在网络上进行交易的过程。举一些例子：假设 A 公司是一家汽车制造商，需要在其汽车上安装微处理器。然后，它将从制造微处理器的 B 公司购买微处理器。另一个例子是当一个人从玩具制造商那里购买一些商品，如玩具。该制造商然后通过网络与包装公司联系，将玩具交付给个人。制造商和包装公司之间的交易是 B2B 交易。B2B 电子商务也涉及一个企业购买另外一家企业的装置或两家企业合并。其中的要点是这种交易必须在网上进行。B2C 的电子商务就是大众群体中的消费者在网上进行购物。在玩具制造商的例子中，个人与玩具制造商之间的购买是 B2C 的交易。

电子商务服务器的模块可以包括用于管理数据和网页、挖掘客户信息、安全执行以及事务管理的模块。电子商务客户端功能可能包括表示管理、用户接口以及缓存数据和托管浏览器。也可能有一个中间层，可以实施业务对象来执行电子商务的业务功能。这些业务功能可能包括经纪、调解、协商、采购、销售、营销和其他电子商务功能。电子商务服务器功能受到网络信息管理技术的影响。除了数据管理功能和业务功能之外，电子商务功能还包括管理分发、异构和联盟的功能。

电子商务还包括非技术方面，如政策、法律、社会影响和心理影响。我们正在以完全不同的方式开展业务，因此我们需要一个模式的转变。如果我们仍然想要传统的购买和销售产品的方式，就不能成功实现电子商务。我们必须更有效率，依靠网络服务和云计算等新兴技术来获得竞争优势。这是因为云计算可以为电子商务获得可扩展的解决方案。电子商务的一些关键组件如图 4.10 所示。

4.5.2　协作与工作流

虽然计算机支持的协同工作（Computer－Supported Cooperative Work，CSCW）的概

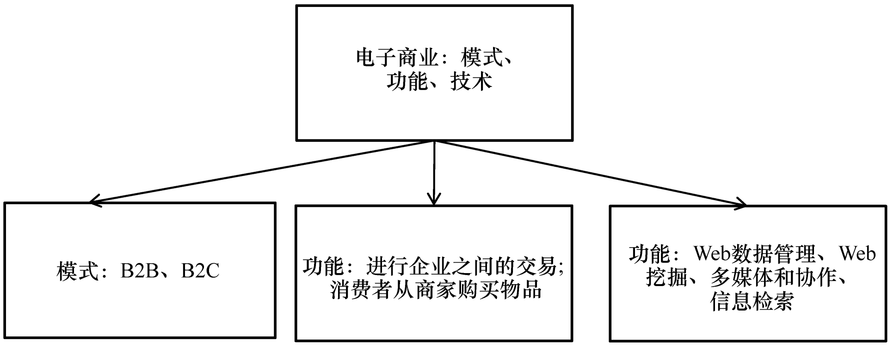


图 4.10　电子商务组件

念在 20 世纪 80 年代初首次提出，但在 20 世纪 90 年代，这个话题才变成热门话题。协作计算使团队、个人和组织能够彼此合作，完成任务或任务的汇集工作。这些任务可能因参与会议、解决具体问题或设计系统而异（参见本章参考文献［ACM 91］）。

数据库社群中对协作计算特别感兴趣的一个方面是工作流计算。工作流被定义为一系列功能的自动化，包括一人或多人执行的数据输入、数据检查和监视等业务流程。一个非常适合工作流自动化过程的例子就是采购流程。一些早期商业的工作流系统产品，主要面向办公环境，基于消息架构。这种架构支持当前工作团队的分布式性质。然而，消息架构通常是基于文件的，并且缺少 DBMS 支持的许多功能，例如数据表示、一致性管理、跟踪和监视。新兴的工作流系统往往要利用数据管理功能。

图 4.11 为 A 组和 B 组正在处理地理问题（如分析和预测北美天气）的示例。这两队必须拥有全球的地图和任何与之相关的注释。一个团队所做的任何更改都应该立即显示给其他团队，两队的沟通就好像他们在同一个房间一样。

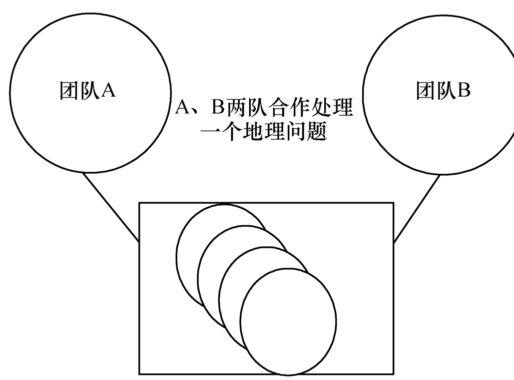


图 4.11　协作计算系统

为了实现这种透明的沟通，需要数据管理的支持。这可以使用 DBMS 来管理数据或某些类型的数据管理器，该数据管理器提供了一些基本功能，如数据完整性、并发访问和检索功能。在上述示例中，数据库可以包括描述团队正在处理的问题的信息、涉及的数据、历史数据以及元数据信息。数据管理器必须提供适当的并发控制功能，以便当两个团队同时访问某一图片并进行更改时，这些更改将协同进行。

网络增加了进一步协作的需要。用户现在可以在网络上共享文档，并在网络上进行论文编写工作和设计工作。企业信息基础设施促进信息和文件的协作和共享。因此，协作工具必须在网络上有效地工作（见本章参考文献［IEEE99］）。如第 5 章所述，多种网络服务标准已经被提出，用于协作和工作流。这样的服务现在正在被云端所提供。

4.5.3 信息集成

图4.12为异构数据库系统或信息源之间的互操作性示例。这个目标是为用户和应用程序提供用于查询和执行交易的透明访问（参见本章参考文献［WIED92］）。请注意，在异构环境中，本地DBMS也可能是异构的。此外，分布式处理器的模块（在4.2.2节中讨论）可以进行DBMS特定处理，以及本地DBMS独立处理。我们称这种分布式处理器为异构分布式处理器（Heterogeneous Distributed Processor，HDP）。要是使这些不同数据库系统之间可以相互操作需要解决几个技术问题。请注意，异构性可能存在于不同的数据模型、模式、查询处理技术、查询语言、事务管理技术、语义、完整性和安全性。

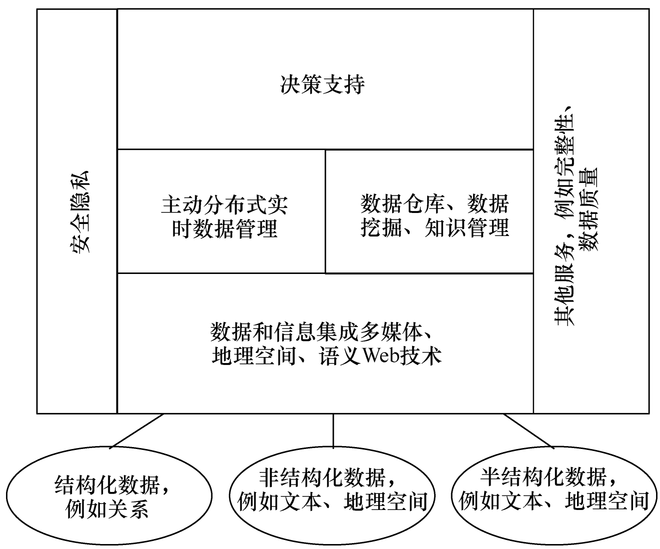


图4.12 信息集成

异构数据库环境中的某些节点可能形成联盟。这样的环境被分类为联盟数据主页环境。如谢思和拉尔森（见本章参考文献［SHET90］）所述，联盟数据库系统是一个合作但自主的数据库系统的集合，它们同属一个联盟团体。也就是说，目标可以是属于联邦的DBMS，彼此合作，并保持一定程度的自主权。第7章将讨论用于信息集成的网络服务。我们将在第六部分讨论一个实验性的安全的基于云的信息集成系统。

4.5.4 信息共享

9/11委员会报告[㊀]鼓励企业从需要知道向需要共享的模式转变。信息共享对于国防和情报机构而言也是重要的，也是医疗机构的重要组成部分。例如，在紧急情况（例如事故）中，患者数据可能必须被共享，以便可以向患者提供最恰当的护理。在正常操作期间，患者数据可能只有在患者授权这样做的情况下才能进行共享。

㊀ 9/11委员会报告为美国恐怖袭击全国委员会报告。——译者注

信息共享增加了安全性和隐私性的复杂性。组织机构必须执行适当的安全和隐私策略，以便只共享合适的数据。此外，还应执行适当策略，以确定在紧急情况下采取行动。信息共享的另一个重要方面是管理信任。例如，联盟中的组织机构是否相互信任？是否应该与另一个不被信任的机构共享信息？是否有不同的信任水平？

也许信息共享的最重要方面是为共享提供激励。即有利于信息共享的策略，在没有鼓励措施时，为什么要共享信息？此外，什么是鼓励措施？它们应该是货币性的还是应该是被认可的奖励，还是为组织机构提供有效执行其职能的工具？

我们正在由空军科学研究办公室（Air Force Office of Scientific Research，AFOSR）资助的多学科大学研究计划（Multi－Disciplinary University Research Initiative，MURI）项目下进行信息共享的广泛研究。有关详细信息，请参阅本章参考文献［FINI09］。图 4. 13 为信息共享的情形。我们的主要重点是应用信息共享策略，以及确定共享的动机。联盟组织之间的信息请求可以被视为网络服务客户端请求，而响应请求的服务器可以被视为网络服务。此外，也可以使用语义 Web 技术，因为数据可以表示为 XML 或 RDF，并且本体可以用于理解在信息共享中的各种概念。我们将在第七部分中讨论我们已经开发的基于云的信息共享保障原型系统。在本书提供的附录 D 中也有信息共享的概述。

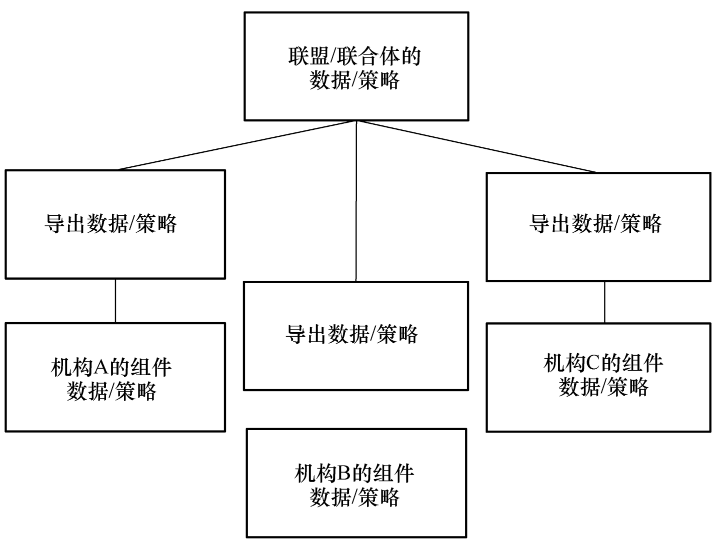


图 4. 13　信息共享

4. 5. 5　社交网络

现在，社交网络已经引起了广泛关注。对于社交网络，主要是研究网络如何形成的。这些社交网络可能是交友网络、恐怖网络、交通网络、通信网络和人类网络。它们通常被统称为社交网络。用于开发和管理社交网络的技术是基于图形理论的技术，数据挖掘技术从个人的行为中提取其社交网络活动，以及挖掘现有网络资源以确定模式，通过可视化技术使网络成员的活动可视化。

通过分析/挖掘网络上的数据，或以其他方式形成社交网络，并确定数据之间的链接。基本上这相当于形成了节点和链接。一旦网络建成，随着新成员进入这个世界，它将不断发展。此外，还要对这些网络进行了分析/挖掘，以提取反馈至网络的有用信息。图 4. 14 说明了这个过程。第四部分讨论了基于云的社交网络实验系统。

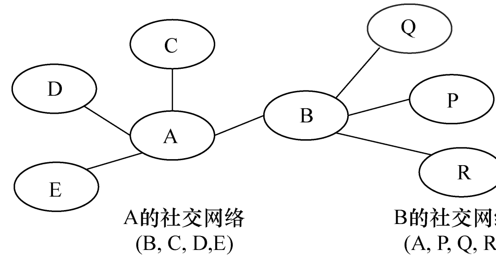


图 4. 14 社交网络

4. 5. 6 供应链管理

最后一个我们需要在讨论中考虑的活动是供应链管理。一些组织机构不可能什么都自己运作，他们需要其他组织机构，以便购买部件或向对方供应部件。例如，考虑一家制造汽车的公司。这样的公司将需要从其他公司获得零件，包括电子设备、发动机和制造汽车所需的其他部件。这些提供商可以从其他提供商处获得其零部件。假设提供商为汽车提供全球定位系统（Global Positioning System，GPS）。该提供商将从其他公司获得其部件（例如，芯片和处理器）。这些组织机构面临的挑战是建立伙伴关系，从而最大限度地获益。他们不想要过时剩余的部件。部件必须在正确的时间到达正确的地方。

IT 在供应链管理中发挥主要作用。数据库系统用于跟踪所有部件和确定它们来自哪里。数据挖掘技术可用于分析数据并确定可供选择的提供商。供应链中的合作伙伴需要信息共享技术来分享信息并最大化其利益。信息集成技术被用于从多个提供商中分离数据库，以便将一个共同的图像呈现给客户。图 4. 15 为供应链管理。供应链管理的网络服务将在第 7 章中讨论。

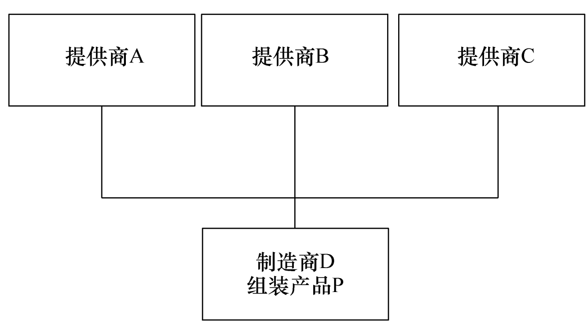


图 4. 15 供应链管理

4. 6 总结和展望

在本章中，我们讨论了数据、信息和知识管理的各个方面。首先，我们讨论了数据管理和复杂的数据管理。然后我们讨论了信息管理和知识管理。特别是，我们讨论了数据模型、数据库系统、多媒体和地理空间数据管理、数据仓库、数据挖掘、信息检索、电子商务和信息共享。基于本章所讨论技术的云计算和云计算安全概念将在第三部分和第五部分给出。在第四部分、第六部分和第七部分，我们将讨论一些我们为数据、信息、知识和活动管理开发的实验性的云计算和云计算安全系统。

数据、信息和知识管理的主要挑战之一是大数据管理和分析。20 年前，大数据被认为是 PB 级的数据，今天则是数据以外的数据。也就是说，大数据是不能用当前技术处理的数据。我们需要用于建模大数据、查询大数据、分析大数据以及可视化大数据的技术。由于存在大量必须被表示、分析和可视化的数据，例如代表社交网络的大量图形结构，我们需要用于管理这种大数据的技术。云计算是管理大数据的重要工具。因此，随后的整本书中的大量论述与大数据管理和分析有关。

参 考 文 献

[ACM91] *Communications of the ACM, Special Issue in Computer Supported Cooperative Work (CSCW)*, 34(12), December 1991.

[BERR97], Berry, M. and G. Linoff, *Data Mining Techniques*, Wiley, Hoboken, NJ, 1997.

[CERI84] Ceri, S. and G. Pelagatti, *Distributed Databases, Principles and Systems*, McGraw-Hill, New York, 1984.

[CODD70] Codd, E. F., A relational model of data for large shared data banks, *Communications of the ACM*, 13, 6, 1970.

[DATE90] Date, C., *An Introduction to Database Systems*, Addison-Wesley, Reading, MA 1990.

[FINI09] Finin, T. et al., Assured information sharing lifecyle, *Proceedings Intelligence and Security Informatics Conference*, Dallas, TX, June 2009.

[GIS] Geospatial Information Systems, http://en.wikipedia.org/wiki/Geographic_information_system

[HARV96] Knowledge management. *Harvard Business Review*, 1996.

[IEEE99] *IEEE Computer, Special Issue in Collaborative Computing Systems*, September 1999.

[IEEE03] Special Issue. *IEEE Compute*, 36(2), January 2003.

[INMO93] Inmon, W., *Building a Data Warehouse*, Wiley, Hoboken, NJ, 1993.

[KDN] KDnuggets, http://www.kdnuggets.com

[MORE01] Morey, D., M. Maybury and B. Thuraisingham (Eds), *Knowledge Management*, MIT Press, Cambridge, MA, 2001.

[SHET90] Sheth, A. and J. Larson, Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), 183–236, 1990.

[SQL3] SQL3, American National Standards Institute, Draft, 1992.

[THUR97] Thuraisingham, B., *Data Management Systems Evolution and Interoperation*, CRC Press, Boca Raton, FL, 1997.

[THUR00] Thuraisingham, B., *Web Data Management and Electronic Commerce*, CRC Press, Boca Raton, FL, 2000.

[THUR01] Thuraisingham, B., *Managing and Mining Multimedia Databases for the Electronic Enterprise*, CRC Press, Boca Raton, FL, 2001.

[THUR02a] Thuraisingham, B., *XML, Databases and the Semantic Web,* CRC Press, Boca Raton, FL, 2001.

[THUR02b] Thuraisingham, B. et al., Collaborative commerce and knowledge management. *Knowledge and Process Management*, 9(1), 43–53, 2002.

[ULLM88] Ullman, J., *Principles of Database and Knowledge Base Systems*, Computer Science Press, New York, NY.

[WIED92] Wiederhold, G., Mediators in the architecture of future information systems. *IEEE Computer*, 25(3), 38–49, 1992.

第一部分总结

第一部分概述了云的支撑技术。特别是，我们讨论了计算的演进、安全技术以及数据、信息和知识管理。我们选择了这些技术，因为它们与本书中讨论的概念、技术、产品和实验系统相关。

第 2 章简要概述了计算的演进。我们讨论了数学逻辑学家的工作、冯 · 诺依曼机器，以及从大型机到个人计算机的计算机演进。最后，我们概述了分布式计算、万维网服务计算和云计算。

第 3 章简要介绍了可靠系统的发展情况。我们讨论了安全系统，包括访问控制中的基本概念，以及自主和强制性策略，安全系统的类型，如安全操作系统、安全数据库、安全网络和新兴技术、网络的影响以及构建安全系统的步骤。接下来，我们讨论了可靠的系统，包括信任、隐私、完整性、质量和实时处理方面。然后，我们更多地关注安全威胁和解决方案的方面。最后，我们提供了使用不可信组件的安全系统的概述。

第 4 章讨论了数据、信息和知识管理的内容。首先，我们讨论了数据管理和复杂的数据管理。然后，我们讨论了信息管理和知识管理。特别是，我们讨论了数据模型、数据库系统、多媒体和地理空间数据管理、数据仓库、数据挖掘、信息检索、电子商务和信息共享。

第二部分　服务安全技术

第二部分简介

Web 服务是云计算的核心。这是因为云向消费者提供基础设施服务、平台服务和应用服务等各种服务。因此，我们将在第二部分讨论服务技术以及服务的安全问题。

第二部分包含 3 章：第 5 章、第 6 章、第 7 章。第 5 章论述 Web 服务以及安全的 Web 服务。首先，论述服务的含义。其次，讨论面向服务的计算的高级概念。接着，我们讨论通过面向服务架构（SOA）和 Web 服务实现的信息系统。最后，论述服务的安全问题。第 6 章阐述语义 Web 服务，这是由于我们云中的一些原型利用了语义 Web 技术。第 7 章以“专用的、安全的 Web 服务”为主题进行论述，例如数据管理服务。

第 5 章　面向服务的计算和安全

5.1　概述

如第 1 章所述，近 60 年来，计算范式已经有很大的发展。起初，计算机被用作数字的处理。后来，它被用来存储和管理数据库中的数据。在数据库为中心的计算模式里，世界被看作是表的合集。再之后，就转移到了面向对象的计算模式。在面向对象的计算模式中，世界是一系列对象的合集。数据库不仅被看作是对象的合集，而且对象也是主要的计算单元。最近，世界已经演变成一系列服务的合集。从根本上来说，任何一个客户都需要从一个服务提供商那里请求一个服务。服务提供商要和客户制定一份协议，即提供服务时，客户为该服务买单。服务可能是医疗保健服务、金融服务或者电信服务等各种服务。这产生了被称为面向服务的计算或服务计算（另见本章参考文献［ZHAN07］和［ERL05］）。换言之，面向服务的计算将世界看作是服务的合集。这些服务由服务提供商所提供，被客户所利用。此外，服务已经成为云计算的心脏和灵魂，因为云计算为客户提供了一系列服务。

面向服务计算的安全性已经成为关键的问题。例如，考虑从代理商订购图书的过程。我们查阅该代理商发布的目录。此时，该代理商必须确保我们有权阅读有关图书的信息（即元数据）。接着可以下订单。如果有必要，该代理商应授予我们相应阅读部分内容的权限。随之书籍中相应的部分将被呈现给我们（客户）阅读。现在这种安全的服务已经能通过软件实现，其过程如下：客户查看代理商的网站，找到书籍并下订单。该网站只会显示客户授权查看的图书。安全的订单管理服务由代理商实现，包括收集订单，发送一条消息给仓库服务并请求该书籍。当仓库服务发现该书在它的清单目录中就发送一条消息给订单管理服务。仓库调用安全服务，并发送书籍中相应的部分给运输服务。然后，运输服务将书籍运送给客户。如果该书籍需以电子的方式显示，那么书籍中相应的部分可能通过订单管理服务进行展示。因此，从订单管理服务、仓库服务和运输服务开始，就需要组成安全的服务。这三项服务为客户提供了他想要的内容。所有这些服务必须加以合适的安全控制。在实施安全服务时，我们需要采取措施，进行访问控制、信任管理和隐私控制。另外，客户得到的文件必须是真实可信的，这就意味着服务必须保持完整性。

服务计算是云计算环境的基本功能，基础设施、平台和软件作为服务由云对外提供。因此，在这一章中我们将论述面向服务的计算和其周边的安全问题，因为它们对安全云计算至关重要。在 5.2 节，我们将给出面向服务计算的概述。服务计算的关键内容是 SOA、Web 服务（WS）和 SOAD。在 5.3 节中，我们将论述面向服务计算的安全问

题。其中包括访问控制、身份管理、一些新兴的网络安全标准以及安全模型。图 5.1 说明了这些概念。本章论述的概念、技术、标准和协议现被用于提供云服务和保障云服务。这些云服务将在本书的第四部分和第六部分被论述。

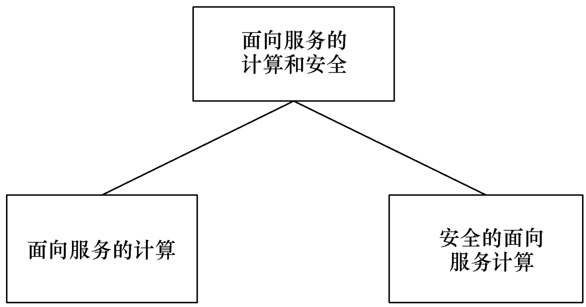


图 5.1　面向服务的计算和安全

5.2　面向服务的计算

5.2.1　服务模式

为了更好地解释服务的概念，我们可以举电信的例子。当我们想使用电话服务时，我们会在 AT&T 或 Sprint 或其他服务提供商中选择一个进行注册。我们也可能会在黄页或者一些报纸的广告中了解它们。无论选定哪一个服务提供商，我们都可以和他们商谈以求获得最好的服务。一旦我们选定了服务提供商，那么服务提供商提出的协议随之而来。如果我们签署了协议，就可以使用服务提供商的电话线路与我们的朋友、亲戚以及商业组织进行电话通信。同样地，对于电子邮件服务来说，服务提供商要么在黄页，要么在网页上发布它的服务。当我们连接到服务提供商，与之签署协议，然后就可使用他们提供的电子邮件服务。在这两种情况下，服务提供商都将以一些我们能够理解的语言发布它们的服务。

现如今，我们正在使用大量的服务，不仅包括电话服务和电子邮件服务，而且还包括医疗保健服务和金融服务等。有了医疗保健服务提供商，我们可以咨询自身的健康情况。金融服务提供商帮我们理财，甚至牧师也可以成为我们的服务提供商，诸如宗教服务。数据和软件也可以作为服务来提供。就数据而言，各种数据中心存储了大量的数据并允许客户将它们用于各种用途。而对软件而言，一些软件公司开发了客户关系管理和医疗保健管理软件，并允许客户以服务的形式使用这些软件。

面向服务的计算对于以实现软件即服务至关重要。一个典型的例子是订购机票。客户打电话给航空预约系统，同代理商对话，代理商负责预订座位并把机票发给客户。有了面向服务计算，客户就可以在线预订机票，系统中的预订服务自查找一个座位，到分配该座位给客户。同时，该服务可以询问客户是否需要预订酒店，并自动向酒店预订服

务发送信息并为客户预订酒店客房。接下来，该服务向租车服务发送一条消息以进行预约，并最终向客户发送消息。航班预订服务、酒店预订服务和预订租车服务组成一个完整的服务，也就是说多种服务可以构成的一个服务。图5.2为这个例子的服务流程。

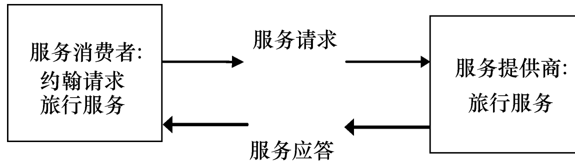


图5.2　面向服务计算的例子

需要注意的是，虽然在面向对象计算中的计算单元是一个对象，但是我们可以把它看成是一个服务。然而，服务的真正实现可能要用到许多程序包或者甚至是多个对象。我们认为面向服务的计算仍处于起步和概念阶段。随着我们在这一领域的进步，适合面向服务计算的语言也许能被开发出来。有必要指出的是，目前面向服务计算中的大多数应用程序都是基于Web的应用程序，特别是与电子商务有关的应用程序。

5.2.1.1　SOA和Web服务

SOA是使用软件技术实现服务的系统架构。在这个体系结构中，有3个主要组件：服务消费者、服务提供商和服务目录。服务提供商以服务目录的标准语言发布其服务。服务消费者请求目录来发现服务。服务目录向服务消费者提供服务提供商的名称或者地址。随之，服务消费者联系服务提供商。许多服务软件都是通过Web技术实现的。因此，实现SOA的服务技术称为WS（Web服务，WS）。服务提供商在一个基于Web的目录上发布它的服务。服务消费者查询这个目录以引导自己找到服务提供商。这种基于Web的目录被称作“通用的描述、发现与集成”（Universal Description，Discovery，and Integration，UDDI）。用于发布服务的语言叫作“Web服务描述语言”（Web Services Description Language，WSDL）。在3个组件之间交换的消息使用称为SOAP的协议。消息传递用XML实现。图5.3为使用WS实现的SOA。

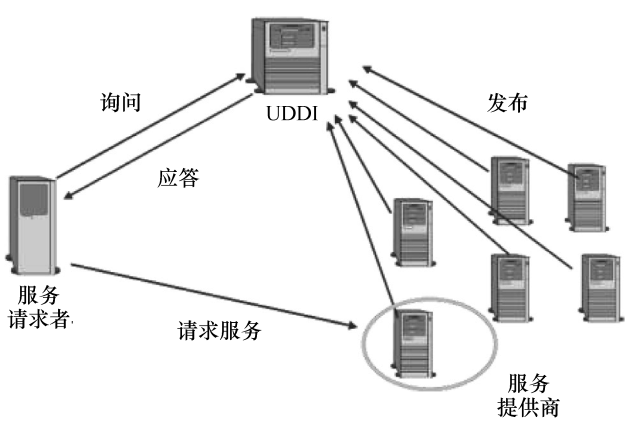


图5.3　SOA和Web服务

5.2.1.2　SOA及其设计

我们都知道，在面向对象的信息系统中，首先需要识别对象、对象类、方法以及对

象之间的关系。此外，还要分析这些对象的活动并纳入到系统的设计中。这种被称为面向对象的分析和设计（OOAD）的方法已经被大家所熟知，UML 也被看作是现在标准的 OOAD 方法。一旦设计了一个系统，接着就要选择适当的对象语言来实现。类似地，在设计面向服务的信息系统之前，我们需要弄清楚需要什么服务，这些服务如何构成，它们如何组织在一起以及这些服务之间有什么样的关系。这种方法被称为面向服务的分析和设计（SOAD）。

相对于 OOAD 的成熟，SOAD 仍处于起步阶段。例如，在我们设计一个实现航班预订功能的信息系统之前，首先，我们需要确定涉及哪些服务？这些服务可能包括航班座位预订、酒店房间预订和出租车预订。接下来，我们要分析这些服务之间的关系。我们可以在航班座位预订的服务页面发出预订酒店房间的请求，也可以请求出租车预订或者再次进行航班座位预订。两个服务可能并行或依次调用。一旦设计好这些服务以及各个服务之间的关系，后面就要制定服务描述。图 5.4 说明了 SOAD 在面向服务的信息系统设计中的应用。

通过 WS 和 SOA 实现的服务将在 5.2.2 节论述，SOAD 将在 5.2.3 节论述。



图 5.4　面向服务的分析与设计

5.2.2　SOA 和 Web 服务

在 5.2.1 节，我们介绍了服务计算的概念（也被认为是面向服务计算），讨论了服务计算的各个方面。在这一小节，我们将通过 SOA 和 WS 来阐述服务计算。SOA 是实现面向服务计算的体系结构。WS 是通过万维网（WWW）实现面向服务计算的一种方式。最流行的面向服务计算的实现方式就是 SOA 和 WS。WS 由 W3C（万维网联盟）和 OASIS 等组织提出的各种标准定义。

在本节中，我们将调研标准化组织关于正在定义的 WS，然后描述 SOA 范式，WS 协议栈以及实现 WS 的替代方法，即 Restful WS。之后再讨论一种流行的 Web 服务技术，它或多或少地导致第一个云得以实现，即亚马逊网络服务［AMAZON］。最后，将论述各种各样云计算中的专业服务。我们会通过本节内容给出几个参考文献，这些参考文献实际上是一些 URL（通用资源定位符），在这些网站中描述各种标准的发展历程。正如我们在本书中强调的一样，WS 技术正在迅速发展。因此，本书的论述可能会随着技术的发展而过时。我们极力主张读者跟上 OASIS 和 W3C 这些标准化组织的发展步伐。这也是我们没有挖掘标准细节的一个主要原因。我们的目标是站在更高的层次去介绍各种概念。

对于 SOA 和 WS 来说，两大主要标准化组织是 W3C 和 OASIS。W3C 已经为 XML 和

安全的 XML 制定了标准，其中包括 XML 加密和 XML 签名技术标准。另外 W3C 也已经为语义化网络、资源描述框架（Resource Description Framework，RDF）、Web 本体语言（Web Ontology Language，OWL）、语义 Web 规则语言（Semantic Web Rule Language，SWRL）以及许多其他相关技术开发出了标准。OASIS 已经为 WS 的认证和授权制定了标准，其中包括安全声明标记语言（Security Assertions Markup Language，SAML）和可扩展访问控制标记语言（eXtensible Access Control Markup Language，XACML）。另外，WS 安全（WS－Security）以及 WS－＊安全框架（WS－＊ Security Framework）是 OASIS 制定的主要安全标准［OASIS］。

另一个重要的标准化组织是 WS 协同组织（Web Services Interoperability，WS－I）。虽然 WS－I 没有制定标准，但他们监督正在开发的标准。还有一个与安全 WS 相关联的团体是自由联盟（Liberty Alliance）。这个组织已经为身份管理提出了标准。一些其他组织，例如对象管理组织（Object Management Group，OMG）和开放地理空间信息联盟（Open Geospatial Consortium，OGC）也已经制定了 WS 的相关标准。图 5. 5 为与 SOA 和 WS 相关的标准化组织。

W3C: 万维网联盟
OGC: 开放地理空间信息联盟
OASIS: 结构化信息标准促进组织

图 5. 5　为服务的标准化组织

正如［OASIS］所定义，WS 指的是允许进行连接的技术，连接在一起的服务是通过使用 WS 实现的。WS 的典型例子是查询服务和目录服务。每个服务都是一条连接的端点。也有一些服务需要底层的计算机系统，它们用来支持提供连接。一个组织内部和外部的服务组合构成一个完整的 SOA。

SOA 支持服务集［ERL05］。服务之间彼此是可以相互通信的。通信可能涉及简单的数据传递，也可能涉及两个或多个服务协调活动，如规划旅行。将服务相互连接需要一些方法。SOA 并不是一种全新的技术。最初的 SOA 可以被认为是分布式组件对象模型（Distributed Component Object Model，DCOM）和基于公共对象请求代理体系结构规范［OMG］（Common Objectrequest Architecture，CORBA ）的对象请求代理（Object Request Brokers，ORB）。如果让 SOA 有效运转，我们需要清楚地了解服务这一术语。服务是一种功能，需要精确定义、自称一体，不依赖于其他服务的上下文或状态。

SOA 有三大主要方面：服务提供商、服务消费者和服务目录。服务提供商在服务目录上发布他的服务。服务消费者去向服务目录请求服务。服务目录回送服务的名字和地址。服务消费者然后发送请求给服务提供商并获得该服务。到目前为止，WS 就是 SOA 实现的最普遍方法。接下来，我们将论述具有 WS 的 SOA 实现的具体技术细节和参数。

5. 2. 2. 1　WS 模型

最初的 Web 模型是基于客户端服务器模式，在这种模式中，Web 客户端通过超文本传输协议（Hypertext Transfer Protocol，HTTP）访问一个 Web 服务器。Web 服务器通常存储一些网页，客户可以请求检索它们。尽管这种模式对于显示网页是胜任的，但是

对于处理站点上的电子商务活动是不足的。因为电子商务中，多个公司必须合作才能去完成一个共同的目标。在这种环境下，多个公司之间可能是异步通信，且每个组织都有可能为另一家组织提供服务。因此，我们需要一个更强而有力的机制去处理电子商务。

在 20 世纪 90 年代后期至 2000 年早期，出现了第二代网络技术，它们的功能不只是以往的网页展示，客户还可以在网站上购买物品。与此同时，面向服务的概念诞生了，电子商务和面向服务计算融合为一体。这造就了 WS 的产生。

WS 技术是最适合实现 SOA 的连接技术。WS 本质上是利用 XML 技术建立一个强连接。一个服务消费者发送一个服务请求给服务提供商。服务提供商返回给消费者一个应答消息。这种请求与随后应答的连接需要以消费者和服务提供商双方都能理解的某种方式来定义。一个服务提供商也可以作为一个服务消费者存在。WSDL 构成了 WS 的基础，它使用 XML 来定义消息。服务提供和使用服务的过程如下：

■ 服务提供商使用 WSDL 描述其服务。这种描述定义被发布到一个服务目录，它可能是 UDDI 目录，也可能是其他形式的目录。

■ 服务消费者向目录提出一个或多个查询来定位一个服务，并确定如何与对应服务通信。

■ 服务提供商所提供的 WSDL 的一部分信息传递给服务消费者。这可以让服务消费者了解这些请求和应答对于服务提供商来说意味着什么。

■ 服务消费者使用 WSDL 向服务提供商发送一个请求。

■ 服务提供商提供给服务消费者一个之前预订的回应。

UDDI 注册表是被最终充当“发现”WS 的手段使用，当然 WS 都是用 WSDL 描述的。我们可以通过各种方式搜索 UDDI 注册表，以获取联系信息和各种组织需要的 WS。UDDI 注册表是一种实现与各种组织目前使用的 WS 保持最新同步状态的方式。能够替代 UDDI 的是电子商务可扩展标记语言（electronic business using eXtensible Markup Language，ebXML）目录。所有的消息都使用 SOAP 发送。（SOAP 一度代表着简单对象访问协议，目前这个首字母缩写词没有这个特别的含义）。SOAP 实际上提供了一个发送 WS 消息的信封。SOAP 通常使用 HTTP 实现，但也不排除使用其他的连接方式。另外，安全和授权认证也是 WS 非常重要的话题。

5.2.2.2 WS 的构成

WS 可以由多个 WS 构成。例如，顾客可能向旅游服务发出请求旅行。然后，旅游服务将请求 3 个服务：航空服务、酒店服务和租车服务。WSDL 将详细说明这些服务并用 UDDI 注册它们。然而，我们也需要一种语言来详细描述服务的流程。为此而开发的语言是业务流程执行语言（Business Process Execution Language，BPEL）。有一种特定的 BPEL 经常被使用，由 IBM 和微软提出，被称为 BPEL4WS（BPEL for WS）。

有了 WS，预订航班、酒店和汽车时，预订次序并不重要。BPEL 声明将以 XML 格式实现，并将发出航班预订、酒店预订和出租车预订的请求。然后，对于任意服务，WSDL将详细描述这些服务的执行操作。如果现在预订航班的次序是最重要的，那么 BPEL 就要建构服务请求的顺序。对于 BPEL 的更多细节，我们可以参考本章参考文献

［SRIV］。图 5.6 为 WS 的构成。

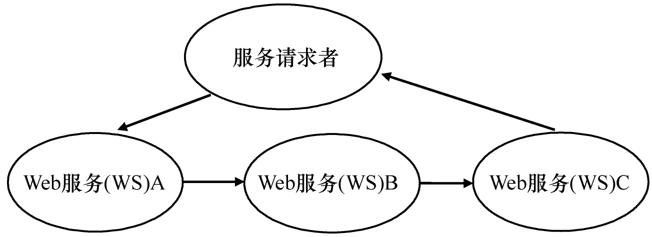


图 5.6　WS 的构成

5.2.2.3　WS 协议

WS 通过协议栈来定义，栈里面包括元数据、消息传递、事务和业务流程、接口、安全、管理和业务领域的相关说明。我们将介绍协议栈的一些重要概念，它们是：元数据、消息传递以及事务和业务流程。图 5.7 说明了 WS 协议栈的构成。

编排/编制协议
服务发现协议：UDDI
服务描述协议：WSDL
消息发送协议：SOAP，XML
传输协议：TCP/IP

图 5.7　Web 服务协议栈

元数据协议包括元数据检索（WS – 元数据交换）、数据服务和消息描述［WSDL］、策略（WS – Policy，WS – Policy Assertions）以及发布和发现［UDDI］，［WSIL］（WS 服务检视语言 Web Service Inspection Language，简写为 WSIL）。在 5.2.2.1 节中，我们论述了 WSDL 和 UDDI。WS 元数据交换实质上给出了其他终端节点去访问 WS 需要获取的元数据。正如在本章参考文献［META］中所描述的那样，“引导与 Web 服务之间的通信，此规范定义了如何将元数据视为资源进行检索，如何将元数据嵌入到 Web 服务端点中引用，以及 Web 服务端点如何选择性支持‘请求 – 响应’交互以用于检索元数据”。WS – Policy 提供了一个策略框架用于指定不同的策略和可选策略。如本章参考文献［POLICY］所述，“WS – Policy 提供了一种灵活可扩展的语法，用于表达基于 XML Web 服务的系统中实体的功能、要求和一般特征。WS – Policy 定义了一个框架和一个模型，用于将这些属性作为策略表达出来。“WS – Policy Assertions” 声明指定了一种语言，通过该语言可以描述策略［PA］。

如本章参考文献［WSIL］所述，WS – Inspection 语言（WSIL）规范提供了一种 XML 格式，用于协助检查可用服务的站点；同时提供了一组规则，用于将这些相关检查信息用于消费。

消息传输协议包括以下方面：消息封装［SOAP］，［MTOM］（消息优化传输机制，Message Transmission Optimization Mechanism）、可靠消息传输（WS – ReliableMessaging，WS 消息可靠传输；WS – Reliability，WS 可靠性）、路由选择或寻址（WS – Addressing，WS 寻址；WS – MessageDelivery，WS 消息交付）、多路消息会话（WS – Enumeration，

WS 枚举；WS – Transfer，WS 转移）、事件和通知（WS – Events，WS 事件；WS – Notification，WS 通知）。在 5.2.2.1 节中，我们论述了 SOAP。MTOM 提供了一种在 WS 之间交换信息的优化机制并且可以与 SOAP 共同使用。WS – Reliable Messaging 是 OASIS 的一个标准，其允许节点之间进行可靠消息的交换（见本章参考文献［RELIABILITY］）。正如在本章参考文献［RELIABILITY］中描述的那样，WS – Reliability 是一种通用和开放的模型，用于确保 WS 的可靠消息传递。本质上讲，WS – Reliability 和 WS – Reliable Messaging 是拥有许多共性并且相互竞争的标准。在本章参考文献［RELIABILITY］中也指出，WS – Reliable Messaging 将取代 WS – Reliability。WS – Addressing 是一种传输中立机制，用来寻址 WS 和识别 WS 节点［ADDRESS］。另一种路由协议是 WS – Message Delivery，它提供了一种“在消息交换模式（Message Exchange Pattern，MEP）的上下文中传递和关联消息的机制，这种机制在服务描述中能够被发现”。为了建立或者使用通信信道，MEP 描述了一种通信协议所需要的消息模式（见本章参考文献［MEP］）。消息交换有两种主要模式：一种是请求应答模式，一种是单向模式。例如，传输控制协议（Transmission Control Protocol，TCP）使用请求应答模式，而用户数据报协议（User Datagram Protocol，UDP）使用单向模式。WS – Enumeration 是一个规范，“它描述如何使一个应用能够从 Web 服务上拥有的数据列表中查询相关内容。因此，WS – Enumeration 对于读取事件日志、消息队列或其他数据集合非常有用”（见本章参考文献［ENUM］）。WS – Transfer 是用 XML 定义来传输 WS – addressable 资源的一种规范，它同样可以创建和删除这些资源（见本章参考文献［ADDRESS］）。WS – Eventing 用于 WS（作为用户）向另一个 Web 服务（可信第三方）注册，以便在某些事件发生时通知它（见本章参考文献［EVENT］）。WS – Notification 是一系列规范的集合，可以向多个 WS 通知各种事件发生情况（见本章参考文献［NOTIFY］）。

用于事务和业务处理的协议包括以下类型：业务流程（BPEL4WS，WS – CDL），事务（WS – Transaction；WS – Coordination：WS 协调；WS – CAF，WS 复合应用框架，Web Services Composite Application Framework）和异步服务（ASAP）。在 5.2.2.2 节中，我们已经论述了 BPEL4WS。WS – CDL 是 Web 服务编排描述语言（Web Services Choreography Description Language，WS – CDL）。在本章参考文献［BPEL］中有这样的阐述：“尽管 BPEL 是一种用于指定参与者在编排中行为的编程语言，它描述了参与者之间的消息交换。所有参与者在编排时都是同等地位的，没有控制中心。”WS – CDL 作为一种基于 XML 的语言，本质上用于描述多方参与者/代理之间的对等协作。WS – Transaction 规定了 WS 域之间事务互操作性的机制（见本章参考文献［TRANS］）。WS – Coordination 描述了一个可扩展框架，用于提供协议来协调分布式应用程序的操作（见本章参考文献［COORD］）。WS – CAF 是由 OASIS 开发的开放框架，以便组合使用包含多个服务的应用程序。此类应用程序称为复合应用程序（见本章参考文献［CAF］）。ASAP（Asynchronous Services Access Protocol，异步服务访问协议）是一种 OASIS 标准，它对 SOAP 进行了扩展，支持通用异步 WS 或长时间运行的 Web 服务（见本章参考文献［ASAP］）。

5.2.2.4　REST

在前面，我们已经讨论了包括 HTTP、SOAP 和 WSDL 在内的 WS 的基本组件。然而，设计软件系统还可以有一种不基于 HTTP 和 WWW 的替代方案，这种方法称为 REST（代表性传输接口）。REST 在其维基百科条目中描述如下（见本章参考文献［REST］）：REST 中的一个重要方面是资源（特定信息的来源）的存在，每个资源都有全局标识符（例如，HTTP 中的 URL）。为了使用这些资源，网络组件（用户代理和源服务器）通过标准化接口（例如，HTTP）进行通信，并交换这些资源的载体（传送信息的实际文档）。

在本章参考文献［REST－SOAP］中也有说明，任何数量的连接器（例如，客户端、服务器、缓存、管道）都可以协调请求，但是每个连接器在协调时都无法“看到过去”自己曾经的请求（这种情况我们称为“分层”，它是 REST 的另一个约束机制，也是其他很多信息和网络架构部分通常采用的做法）。因此，应用程序可以通过了解两件事情做到与资源进行交互：资源的标识符和所需的动作——它不需要知道是否存在缓存、代理、网关、防火墙、管道或其他与实际持有信息的服务器之间的任何东西。然而，该应用程序需要了解返回信息（载体）的格式（例如，HTML、XML 文档）。

5.2.3　SOAD

分析和设计方法对于良好的软件开发至关重要。最初为了实现更好的软件工程，我们有软件分析和设计方法。伴随着像 Smalltalk、C＋＋和 Java 等这些面向对象软件开发工具的爆炸式发展，出现了一些面向对象的分析和设计方法，也称为 OOAD。经过大量争论之后，最终 UML 成为 OOAD 的统一方法。现今面向服务的计算正在迅猛发展。与面向对象编程不同，面向服务的编程没有自己的语言。我们目前可以使用 C、C＋＋和 Java 等语言进行面向服务的计算。然而，主要的问题是我们如何对这些服务进行建模和分析？是否有统一的 SOAD 方法？目前，已经有人提出了几种方法，我们相信，最终会有统一的方法。

在定义软件生命周期以及对象的生命周期方面，人们已经做了很多努力。例如，在软件方面，第一步是收集需求，然后确定输入/输出，接下来设计算法，然后开发软件，测试软件，将软件集成到系统中，进行系统测试，以及最后部署软件。然而这个过程并没有结束，我们还必须维护软件，修复错误，并向客户提供技术支持。这也被称为是自上而下的软件开发方法。在自下而上的方法中，则是根据需要开发软件模块。类似的，对于对象软件开发，在自上而下的方法中，必须分析应用程序，确定对象以及对象之间的关系，开发对象，测试对象和集成对象。系统也必须得到维护。在自下而上的方法中，是根据需要开发对象。

同样，服务也有生命周期。Thomas Erl（托马斯·伊尔）在他的 SOA 书籍中解释了服务生命周期（见本章参考文献［ERL05］）。他提出了 3 种开展服务的方式：一是自上而下的方法，二是自下而上的方法，三是他所说的敏捷方法。在自上而下的方法中，首先必须进行分析，然后设计服务、开发服务、测试服务、整合服务，最后维护服务。

在自下而上的方法中，服务根据需求进行设计和开发。在敏捷方法中，采用综合方法。也就是说，分析应用程序并确定服务。但是，不必等待所有服务都确定下来才去开发。我们可以先开发一些关键服务，然后进行更多的分析和设计，随之开发一些其他服务。该过程会持续下去并适应于应用程序的变化。图 5.8 描述了服务生命周期。

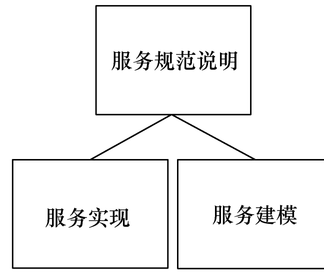


图 5.8　面向服务的生命周期

通常，我们会在 SOA 类中讨论面向对象的分析方法和 UML。虽然我们可以学习 OOAD 中内在的一些设计原则，但 SOAD 与 OOAD 还是不一样，毕竟 SOAD 是用于设计服务的。需要注意的是，对象可能用于实现服务。但是，服务的概念与对象的概念是不一样的。因此，如果想要设计出良好的基于服务的系统，重要的是要了解 SOAD 的概念。

第一步是分析应用程序并确定描述应用程序的服务。首先，必须确定由每个服务封装的逻辑、逻辑的复用以及该服务的接口。另外，服务也应该具有自治性。下一步，需要确定的是包括服务组合在内的服务之间的关系。在自上而下的方法中，必须确定所有服务及其之间的关系，才能进行服务的详细设计和开发。对于大型应用设计，这或许是不可行的。在自下而上设计方法中，可以确定部分服务再开始开发。在敏捷设计中，这两种策略融为一体。在航空预订应用中，这些服务是预订航班、预订酒店和预订出租车。它们可以作为 3 个独立的服务来实现，也可以由预订航班服务来调用预订酒店和预订出租车服务。

Erl 在他的书（见本章参考文献［ERL05］）中，为商业服务提供了一个强有力的案例。在这个案例中，业务逻辑被建模为服务。他进一步指出，这种方法说明基于业务流程的 SOA 实现的各个阶段。这种业务流程在根本上实现了使不同的应用程序能够交互的工作流程逻辑。还应指出的是，业务流程本身可以被实现为服务。因此，业务流程服务可以被不同的应用去调用，而这些应用以服务的形式出现，可以彼此进行交互操作。这种商业服务模式也能促进服务的再利用。例如，应付账款服务可能会被不同的应用程序重复使用。

这里的主要问题是你该如何定义一个服务？从最高层面上看，完整的应用例如订单管理可以是一项服务。然而，这不是我们所希望的。在另一极端情况下，业务流程可以分为若干个步骤，并且每个步骤都可以是一个服务。以订单管理为例子，这些步骤包括①在网上书店搜索你想要的书籍，②比较这本书在各个不同书店的价格，③审查运输规则和退货策略，④检查书店是否支持你拥有的信用卡进行消费，⑤选择一个书店，⑥搜索这本书，⑦将书放在购物车中，⑧填写全部细节后，购买该书，⑨等待确认，⑩付款离开。现在每个步骤都可以是一个服务。这意味着一个中型应用程序，可能有数百种服务。因此，我们需要对步骤进行成组，将一些相关任务集成在一个服务中，以减少服务个数。正如 Erl 所说的那样，在良好的服务设计中，有两大设计原则（见本章参考文献

［ERL05］）：一个是可重用性，另一个是自治性。服务也可以根据要执行的操作来定义。在开始阶段我们不能明确定义的服务或操作，被称为候选服务和候选操作。

接下来，审查候选服务并确定它们之间的关系。一个服务可能会调用其他服务。两个服务可以结合成一个复合服务。这意味着要识别两个服务的边界和接口，尽量使服务的组合和分离耦合度低（尽可能清晰）。如果服务互相依赖可能导致服务设计变得复杂。服务操作可以是简单的操作，例如进行计算；也可以是复杂操作，比如调用多个服务。

一旦候选服务和服务操作得到确认，下一步就是对服务和服务操作进行设计，并提炼新的候选服务。需要注意的是，这也取决于是否遵循自上向下、自下而上或敏捷策略。不同的设计策略会导致不同的情形，或者所有服务均要在开发之前被定义下来，或者可以先定义一些服务，然后再开始开发，而其他服务仍然处于定义之中。本章参考文献［ERL05］给出了 SOAD 的各种方法的细节，这些方法与服务型企业和企业建模相关。我们将在下一节进一步研究相关方法。

在 20 世纪 80 年代末和 90 年代初期，几种面向对象的编辑方法处于相互竞争之中。这些方法包括 Rumbaugh 的 OMT 以及 Booch 和 Ivar Jacobson 的用例法。之后，在 UML 影响下，各种方法实现了统一，目前已经形成了一个 OOAD 标准。但是，我们对于 SOAD 还没有一些标准。针对目前存在的多个方法，当我们更好地了解 SOAD 方法时，我们期望可以将各种方法统一起来。目前存在的一些方法中包括 IBM 的 SOAD，它被称为 SOMA 和 UML 服务。我们在本节论述的这些方法，图 5.9 进行了相关说明。在本章参考文献［MODEL］、［SURVEY］和［SOAD］中详细论述了面向服务的建模。下面我们将讨论一些知名的 SOAD 方法。更多详细信息请参见本章参考文献［THUR10］。

SOMA：面向服务的建模架构
SOMF：面向服务的建模框架
服务型UML

图 5.9　SOAD 的方法

5.2.3.1　IBM SOAD

IBM 公司首先提出了 SOAD 这一术语，然后用 SOMA 说明它。正如在 Ramollari 等人的调查报告中关于 SOAD 方法所述，IBM 的 SOAD 所提出的要素应该是 SOAD 方法的一部分。因此，这是一个抽象框架而不是一个完整的方法（见本章参考文献［IBM］）。SOAD 建立在现有的成熟技术之上，如 OOAD。另外，它还引入了 SOA 特定技术，例如服务概念化、服务分类和聚合。

IBM SOMA：IBM 的 SOMA（面向服务的建模架构）可以被认为是 IBM 的 SOAD 的一个具体实现。如本章参考文献［SOMA］的描述，SOMA 通过服务的识别，规格细化和实现，以及实现服务组件和用于构成服务流程的组件来实施 SOAD。IBM 的方法扩展了面向对象的基于组件的分析和 SOA 设计方法。这也说明，SOMA 通过信息补偿技术确认服务、组件边界、流程、组成情况、信息，这种技术包括域分解、目标服务建模和现

有资产分析。

5.2.3.2 面向服务的建模框架

另一个 SOAD 方法是 SOMF（面向服务的建模框架）。如本章参考文献［SOAD］中所述，SOMF 是一种以面向服务开发的生命周期方法，并且支持许多建模实践和原型，有助于成功地进行面向服务的生命周期管理和建模。建模有四个部分：实践、环境、原型和工件。

5.2.3.3 服务型 UML

那些已经使用过 UML 的人大力推广在 SOAD 中应用 UML。IBM 的 UML 可视化建模软件（Rational Rose），其中有一部分就是 UML 到 SOA 转换的工具。如 IBM［IBM］所述，UML 到 SOA 的转换通常接受 UML 模型作为其源，并创建特定域的 SOA 输出。

SOAD 的大部分工作都受到 OOAD 的影响，OOAD 也包括基于 UML 的建模。SOAD 还处于起步阶段。因此，我们相信就像 UML 在 OOAD 领域赢得战斗一样，在 SOAD 领域也很可能会有一个统一的方法。关于 UML 的详细信息可以在本章参考文献［UML］中找到。

5.3 面向服务计算的安全

5.3.1 服务范式的安全

服务的安全性本质上要求将安全性纳入到服务技术中。例如，代理必须根据哪些证书来调用 Web 服务？Web 服务必须使用哪些凭证来调用另一个 Web 服务？是否每个服务描述对代理都可见？如何在 Web 服务描述上执行访问控制？安全性如何纳入 SOA？什么是 W3C 和 OASIS 提出的安全标准？我们将探索这些问题的答案。有关安全服务的更多详细信息，请参阅本章参考文献［BERT06］。最近，出版了一本收录有关 WS 安全的论文集，本书对这个领域的新兴标准和研究方向提供了一个很好的概述（见本章参考文献［GUTI10］）。

为了更好地说明服务安全的概念，我们将使用信用申请的示例。假设我们要收到自己的信用报告。我们会联系那个可以提供信用报告的服务提供商。首先，我们应该有机会读取相应服务提供商的信息。一旦确认这个服务提供商，我们将向服务提供商发出请求。服务提供商应确保我们有权访问相应的特定服务，另外，我们有时需要读取有关其检索信用的信息。为了做到这一点，我们还必须向服务提供商发送一些相关身份信息。如果服务不安全，那么任何人都可以获得其他人的信用报告。同样，为了获得医疗保健报告，服务提供商应确保请求服务的人员具有阅读医疗保健记录的相应凭证。此外，医疗保健记录的拥有者可以执行各种隐私策略，在这种情况下，服务提供商应仅向消费者发布其相应的其他信息。在某些情况下，消费者可以通过服务提供商来购买信息。另外，服务提供商通常会说明其保护隐私的策略，如果消费者同意这些策略，则可以发布关于他/她的私人信息。

这个简单的例子说明了以下几个方面。一是该服务的用户必须由服务提供商证实。服务提供商必须可以信任，意思就是用户不想从不可靠的提供商那里获得服务。二是服务提供商必须确保用户/消费者具有真实的证书来获取服务，并且发布的任何信息都是消费者有权阅读的信息。三是服务提供商还必须确保关于其他人的私人信息不能发布给消费者。因此，从根本上来说，我们需要保证 WS 具有机密性、隐私、信任和完整性特征。

实现 SOA 的关键是利用 WS。因此，需要通过安全的 WS 来实现安全的 SOA。基础的 SOA 本质上描述消费者从 UDDI 请求服务，UDDI 发送服务的名称/地址，然后，消费者从服务提供商获得该服务。使用安全的 SOA，我们必须确保消费者、UDDI 和服务提供商之间的通信是安全的。此外，只有被授权的消费者才能获得所需的服务，并且以 XML 编码的 SOAP 消息必须是安全的。XML 通过加密标准提供机密性，通过签名标准提供完整性。XML 的加密和签名都是由 W3C 提供标准。

WS 的安全和授权规范基于 XML，可以在本章参考文献［OASIS］、［XACML］和［SAML］中找到相关说明。这些文献提出了各种类型的控制，包括访问控制、权限、声明和保护，我们将在下一节对其中的一些控制进行论述。相关规范清单包括以下内容：

- 可扩展访问控制标记语言（eXtensible Access Control Markup Language，XACML）
- 可扩展权限标记语言（eXtensible Rights Markup Language，XRML）
- 安全声明标记语（Security Assertion Markup Language，SAML）
- 服务保护标记语言（Service Protection Markup Language，SPML）
- Web 服务安全（Web Services Security，WSS）
- XML 通用生物识别格式（XML Common Biometric Format，XCBF）
- XML 密钥管理规范（XML Key Management Specification，XKMS）

OASIS 是推动 WS 安全标准的关键标准组织。这是一个非营利性的全球联盟，它推动电子商务标准的发展、融合和采用。OASIS 提供的两个著名标准是 XACML 和 SAML。XACML 提供对授权活动的细粒度控制、访问请求者的特征作用、请求所用的协议、基于活动类的授权和内容自检。SAML 是用于交换认证和授权信息的 XML 框架。至于 WS 安全的细节，我们将在后面的部分讨论。

我们是第一个根据 OMT 模型检查 OOAD 的安全性的。开发了一个安全的对象模型、安全的动态模型和安全的功能模型。从那以后，几位研究人员相继开发出基于对象的 OOAD 安全方法。通过这个 SOAD 方法，能够达到确认应用程序中的服务以及服务之间的关系的目标。例如，图书订单应用程序的服务将包括订单保存服务、仓库服务和送货服务。这些服务与它们的各种安全策略相关联。我们面临的挑战是如何以适当的建模语言定义服务和策略。

目前，关于安全的面向服务的设计和分析（Secure Service - Oriented Design and Analyses，S - SOAD）上，几乎还没有工作。本章后半部分，我们将根据安全的 OOAD 的发展情况进行尝试性论述。也就是说，我们将研究 SOAD 规则，审查 SOAD 的安全性。应

该指出的是随着 WS 的安全性以及 SOAD 方法的成熟，我们将看到更好的 S－SOAD 方法。

身份管理（通常也称为联合身份管理）与 WS 密切相关。在访问资源之前，用户以及 WS 必须进行身份验证。单点登录（Single Sign－On，SSO）是一个流行的解决方案，一次性登录可以让用户或服务访问各种资源。此外，SAML 目前为 WS 提供身份验证设施。然而，随着电子商务的监管要求的变化，我们需要一个更强大的认证机制，这种机制已经被称为身份管理。

如本章参考文献［FED］中所述，联合身份“描述了技术、标准和用例，以便在其他自主安全域中实现身份信息的可移植性。”使用它的目标是确保一个域的用户以无缝的方式利用另一个域的所有技术。需要注意的是，联盟是关于协作执行任务（例如 B2B 操作）或解决特定问题的组织。虽然建立联盟的想法已经存在了许多年，但只有最近随着 WS 标准的出现才使得我们能够形成现实的联盟。在这样的联盟中，必须对用户访问资源进行管理，且不会对用户造成负担。

服务的安全标准基本上由 W3C 和 Security SIS 开发。本质上，Web Services 1.0 主要是服务消费者请求服务提供商提供服务，然后服务提供商提供服务。在 SOAP 中交换的 XML 消息被加密和签名，以保证机密性和完整性。我们的目的是加密消息以提供机密性并签署消息以确保消息不被篡改。XML 密钥管理和 XML 加密在提供信息的保密性和完整性方面发挥了重要作用。

针对 Web Services 2.0 已经形成了几个附加的标准，包括安全消息传输、可靠性和身份管理。另外，策略的标准管理如 WS－Policy，访问控制标准如 XACML，以及诸如 SAML 等安全声明的标准也已经形成。我们会在本章后面讨论这些标准。

对于许多应用程序，只有访问控制模型是不够的。例如，在复合 WS 的情况下，一个 Web 服务 S1 可以调用另一个 Web 服务 S2。在这样的调用中，S1 的特权将被执行，但这不是那些调用 S1 的用户 U 一定要执行的。这意味着返回给 U 的信息可能是用户未被授权可以知道的内容。为了避免这种安全泄露，用户 U 必须将其权限委派给 S1，以便 U 的权限在 S1 调用 S2 时使用。这种调用由所使用的授权模型（见本章参考文献［SHE08］）来管理。

复合 WS 的另一个安全问题是信息流。当 WS 被组合时，至关重要的是不存在从高层流到低层的信息流。我们重点研究关于 WS 安全的多个方面，包括 Web 服务组合的授权模型和信息流。图 5.10 中举例说明了安全的面向服务计算的概念。

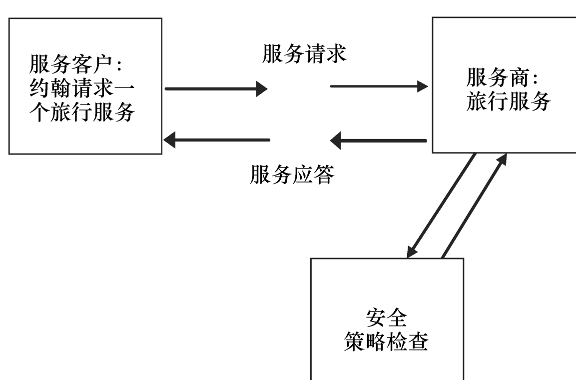


图 5.10　安全的面向服务计算例子

5.3.2　安全的 SOA 和 WS

我们的方法是通过 WS 实现 SOA。因此，SOA 安全本质上是关于 WS 的安全。现在已经有为 Web Services 1.0 提供安全性的规范，它们是 WS – Security、XML – Signature 和 XML – Encryption。WS – * Security 是关于 SOA 安全性的第二代技术。SSO 是一种安全机制的集中形式，它对 WS – Security 进行了扩展。SOA 安全性的相关规范包括：WS – Security、WS – SecurityPolicy、WS – Trust、WS – SecureConversation（WS 安全对话）、WS – Federation、XACML、可扩展权限标记语言（Extensible Rights Markup Language）、XML 密钥管理、XML、签名、SAML、. NET Passport、安全套接字层和 WS – I Basic Security Profile（WS – I 基本安全配置文件）。图 5.11 为安全 WS 架构的概念。

接下来，我们将对 WS – Security 和 WS * – Security 进行简要介绍。关于 SOA 安全的具体细节，请参考本章参考文献［BERT06］、［WSS］。

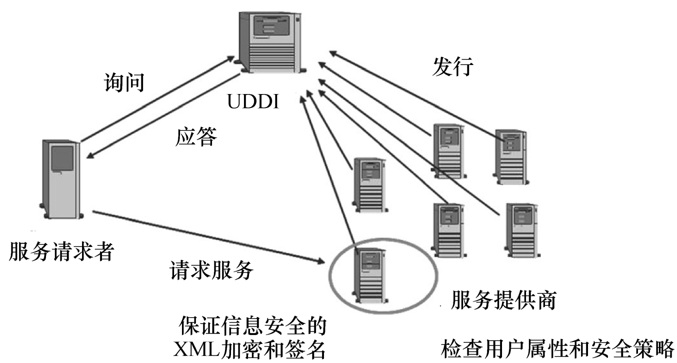


图 5.11　安全的 SOA 和 Web 服务

5.3.2.1　WS – Security

在介绍 WS – Security 的细节之前，我们将讨论 WS 需要的一些安全属性。它们包括以下内容：

鉴别：对于服务请求者访问安全服务提供商，首先必须提供说明其来源或拥有者的信息，这里指的是要做出声明。

验证：传递给接收者的消息必须经证明该消息实际上来自其声明的发送者。

授权：一旦认证，消息的接收者需要决定允许请求者做什么。

单点登录：用户必须单点登录一次后才能访问所有资源。它由 SAML、. NET Passport 和 XACML 支持。

机密性和完整性：机密性是关心消息内容的隐私保护；完整性是确保消息不被改变。

传输级别和消息级别的安全性：SSL（安全 HTTP）提供传输安全性；XML 加密和 XML 签名提供保证消息级别的机密性和完整性。

WS 安全主要是指提供确保消息的完整性和机密性的设施，并确保该服务仅对策略

表达式的消息请求服务。标准的 WS 安全包括提供 Web 服务安全框架，这也是 Web 服务架构的组成部分。接下来，我们将简略描述 WS－Security 的各个组件。

XML 加密：XML 加密语法和处理过程是一种 W3C 标准，并早在 2002 年推广使用。其目的是为应该用程序交换结构化数据时提供机密性。主要通过表示成标准的数字加密资源，将加密信息从加密数据中分离出来，并支持从加密数据中处理加密信息的参考机制（反之亦然），提供用于将加密密钥信息传递给接收者并提供对 XML 文档的一部分或全部加密的机制等措施来实现。

XML 签名：这也是 W3C 标准，并在 2002 年推出。XML 签名是许多 WS 安全标准（例如 XKMS 和 WS－Security）的组成部分。其目标是将数字签名表示为 XML 元素和用于创建此 XML 元素的方法规则。签名的数据项可以是不同的类型和粒度（XML 文档、XML 元素、也包括任何类型的数字数据的文件）。

保证 SOAP 消息安全对于 WS－Security 至关重要。SOAP 消息安全性 1.0 版本于 2006 年成为被批准的 OASIS 标准规范。其目标是通过使用现有的数字签名、加密和安全令牌机制来提供单一的 SOAP 消息完整性和保密性，并提供将安全令牌与消息内容（标题和主体）相关联的机制和支持可扩展性（即，支持多个安全令牌格式）。安全令牌是与安全相关的一种信息表示（例如，9.509 证书、Kerberos 票证和认证者，来自 SIM 卡的移动设备安全令牌、用户名等）。署名的安全令牌也是一种安全令牌，其中包含一组由发行人加密认证的相关声明（断言）（示例：9.509 证书和 Kerberos 票证）。

现在我们回到 WS－Security 上。WS－Security 是什么？它通过消息完整性，消息保密性和单消息认证等增强了 SOAP 消息传递，从而提供了质量保护。这些机制也可用于容纳各种安全模型和加密技术。WS－Security 还提供了一种通用的可扩展机制，用于将安全令牌与消息相关联。此外，WS－Security 还描述了如何编码二进制安全令牌（9.509 证书和 Kerberos 票证）。图 5.12 说明了 WS－Security 的构成。

WS-Security： 签署SOAP消息，加密SOAP消息和附加的安全令牌机制
SOAP基础
XML安全性：XML签名，XML加密
传输级安全性：SSL / TLS
网络级安全性：IPSec

图 5.12　WS－Security

5.3.2.2　WS－＊Security

WS－＊Security 标准规范解决了互操作性方面的安全问题。要说明的是，每个标准规范也都有一部分描述专门说明自身没有解决的安全威胁。WS－＊Security 框架使用

WS－Security。该框架已经在 Microsoft .NET Framework 2.0（WSE3.0）、SUN Web 服务互操作技术（Web Services Interoperability Technologg，WSIT）、IBM WebSphere 和开源软件（见本章参考文献［APAC］）中实施。在理论上，框架要求使用分层的方法，其中每个上层标准可以复用和扩展下层标准的规范。实际上，不同组织发布的规范并不总是兼容的。但是，他们坚持使用配置文件，提高互操作性。应该注意的是，不同提供商的实现并不总是可互操作的。WS－＊Security 中，提供安全性的 3 个主要组件是：WS－Policy、WS－Trust 和 WS－Addressing。WS－Addressing 是传输中立机制的规范，允许 Web 服务传送地址信息。下面我们将讨论 WS－Policy 和 WS－Trust。图 5.13 说明了 WS－＊Security 的构成情况。注意，这个堆栈结构实际是不断发展变化的。其中一些标准已经被采纳，而其他标准正在实验阶段，还有一些只在规范阶段。因此，一个特定协议所处的状态可能随时间而变化。

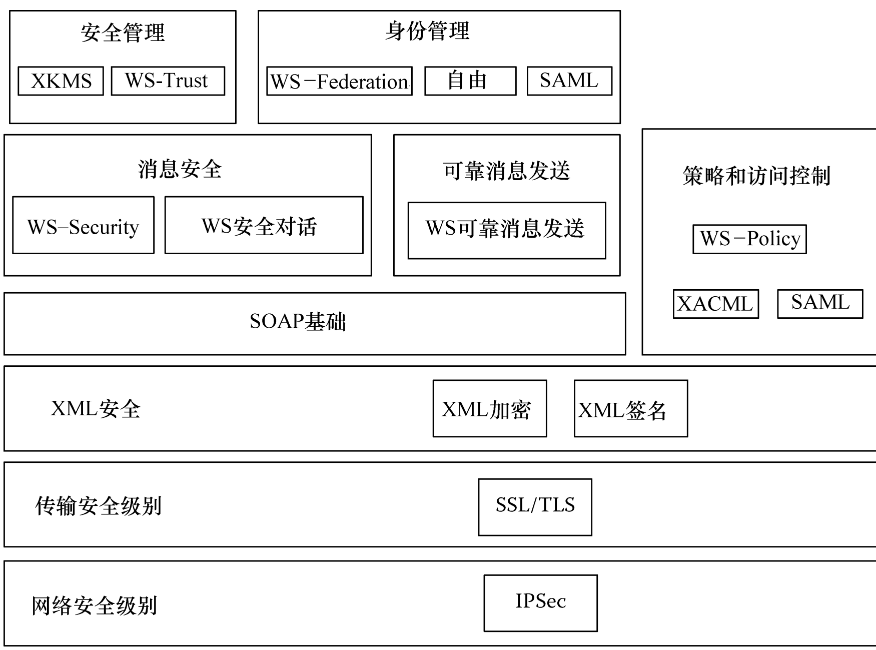


图 5.13　WS－＊Security

WS－Policy：Web 服务策略 1.2 框架（WS－Policy）是一种 W3C 提交的标准。一个策略是潜在的关于待选策略的空集。待选策略是没有顺序的。待选策略是潜在的关于策略声明空集。没有声明的待选策略表示没有行为发出者。待选策略也是相互排斥的（也就是异或关系）。策略声明能识别策略主体的要求（或能力），可以表示特定域（例如，安全性，事务）的语义，并且应该在独立、特定域规范中被定义。

WS－Policy 被认为是一个可扩展的模型，它能够表示所有类型的特定域策略模型：传输级安全性，资源使用策略，甚至是端到端的业务流程级别策略。它定义了一个基本的策略、策略描述和策略声明模型。WS－Policy 还可以并入其他策略模型，如 SAML 和

XACML。WS - Policy Assertion 定义了一些通用的策略声明。WS - Policy 附件定义了策略如何与服务相关联，或者将策略直接嵌入 WSDL 定义中，或间接通过 UDDI 关联策略。WS - SecurityPolicy 定义了与 WS - Security 定义的安全说明相对应的安全策略声明：消息完整性声明，消息认证声明和消息安全性令牌声明。

WS - Policy 和 WS - PolicyAttachment 的目标是为了提供一些机制，来表达 WS 作为策略的能力和要求。WS - Policy 的策略观点如下：一种策略用于表达两个 Web 服务端点之间交互的条件。也就是，Web 服务提供商公开一个策略表达它提供服务的条件。请求者可能会使用这个策略去决定是否使用该服务。

WS - Trust：如本章参考文献［TRUST］所述，WS - Trust 是一个 WS - ＊Specification 和 OASIS 提供 WS - Security 扩展的标准。它涉及发布、更新和验证安全令牌。同时，它也作为经纪人通过安全对话，协调参与者之间利用安全的信息交换维持信任关系。安全性（机密性和完整性）往往通过加密、数字签名和证书来实现。归根结底，安全性要依赖加密密钥和安全令牌的安全管理，包括：密钥/安全令牌发布、密钥/安全令牌传输、密钥/安全令牌存储以及密钥/安全令牌交换。Web 服务信任语言（Web Services Trust Language，简称 WS - Trust）于 2005 年发布，其目标是在不同的信任域之间发布和传递证书。WS - Trust 对 WS - Security 进行了扩展，提供用于发布、更新和验证安全令牌以及建立安全令牌的方法，评估信任关系的存在和协调信任关系的方法。对 WS - Security 保护的 SOAP 消息的接收者来说，有 3 个潜在的包含在安全令牌头部的问题要解决。一是格式：令牌的格式或语法对于接收者来说是未知的；二是信任：接收方可能无法建立信任链，这种信任链是从其自己的信任锚［例如，其 X. 509 证书管理机构，本地 Kerberos KDC（密钥分发中心）］或 SAML 管理中心到发行商或令牌签名者；三是命名空间：由于语法差异，接收者可能无法直接理解令牌内的一些声明。

消息可靠性由 WS - ReliableMessaging 标准提出。而消息安全性由 WS - Security 和 SecureConversation 标准提出。如本章参考文献［CONV］所述，WS - SecureConversation 是由 IBM 等创建的 Web 服务规范，与 WS - Security、WS - Trust 和 WS - Policy 协同工作，允许创建和共享安全上下文。实际上，WS - SecureConversation 的目标是为多个 SOAP 消息交换建立安全上下文，而这反过来减少了密钥建立的开销。对话则主要集中在 Web 服务参与者参与的公共过程。WSCL（Web Services Conversation Language）是 Web 服务对话语言。更准确地说，WS - Conversation 提供了跨越一个或多个消息的安全通信，并扩展了 WS - Security 机制。为了增加安全性，对消息对话需要施加一系列派生密钥产生安全上下文，在安全上下文的两个端点间再建立和共享一系列 SOAP 消息（会话），这些步骤也减慢了认证过程。安全上下文被定义为一个新令牌类型，它通过绑定 WS - Trust 来获取。它也是一种抽象概念，指的是已经建立的认证状态和可能具有其他相关安全性属性的协商密钥。安全上下文令牌（Security Context Token，SCT）是该安全上下文抽象概念的表示，它允许上下文由 URI 命名并与 WS - Security 一起使用。

策略和访问控制由 WS - Policy、XACML 和 SAML 提出。SAML 由 OASIS 的基于 XML 安全服务技术委员会（Security Services Technical Committee，SSTC）开发，其主要

目标是提供认证和授权，能够促进不同身份验证和授权系统之间的互操作性。它通过定义基于 XML 的框架来实现这一点，以便在使用不同安全基础设施，例如，Public Key Infrastructure（PKI）、Kerberos、LDAP 等的计算实体之间传达安全性和身份信息（例如，身份验证、授权和属性）。XACML 2.0 版也是 OASIS 标准，它是一种用于管理资源访问的通用访问控制策略语言，描述了策略语言和访问控制决策请求/响应语言。另外，它还提供基于主体和客体属性的细粒度访问控制。XACML 2.0 基于 SAML 建立，并与其一致。

安全管理基本上由 SAML 和 XKMS 完成。如 W3C 所述，XML 密钥管理规范（XML Key Management Specifcation，XKMS）包括两个部分：XML 密钥信息服务规范（XML Key Information Service Specifcation，XKISS）和 XML 密钥注册服务规范（XML Key Registration Service Specifcation，XKRSS）。在 W3C 规范中，XKISS 允许客户端将处理 XML 签名元素所需的部分或全部任务委托给 XKMS 服务。本质上，XKISS 通过成为 XKMS 服务的客户端，最大限度地减少使用 XML 签名的应用程序的复杂性。通过这种方式，W3C 说明了该应用程序可以减少基于 PKI 建立信任关系的复杂性和过多语法。W3C 也说明了，XKRSS 描述的是一种用于公钥信息注册和后续管理的协议。我们最后要讨论的组件是身份管理，它的标准提供者是 SAML、WS – Federation 和自由联盟。

如本章参考文献［FED］所述，WS – Federation 是由 BEA Systems（现在的 Oracle）、IBM、Microsoft 和其他公司开发的身份联合规范。这个规范定义了一种机制，允许不同安全实体协调关于身份、身份属性和验证等信息。这个自由联盟成立于 2001 年 9 月，由大约 30 个组织组成，目的是建立身份管理的开放标准、准则，并提供最佳范例。

5.3.3 安全的 SOAD

随着服务技术的大量出现，我们需要一种基于服务并有效地为应用程序建模的方法。为此我们开发了 SOAD 方法，而 IBM 则是该领域的领导者之一。在本章参考文献［THUR10］中，我们讨论了面向服务的生命周期以及包括 SOMA 和 SOMF 在内的 SOAD 的方法。虽然 SOAD 适用于服务建模，但为了建立安全的服务模型，我们需要安全的 SOAD。在本节中，我们将讨论开发安全的 SOAD 的一些初步想法。

安全的服务建模从 OOAD 中获益良多。OOAD 方法在 20 世纪 80 年代和 90 年代被开发，并从实体关系建模演变而来。这些方法包括 Rumbaugh 的 OMT 和 Booch 的类图。我们在本章参考文献［SELL93］中将安全性纳入 OMT。例如，我们开发了一种从动态和功能两个角度对对象之间的关系进行建模的方法，还应用这个方法到医疗应用以及实时应用领域（见本章参考文献［THUR94a］、［THUR94b］）。

正如我们在本章参考文献［THUR10］中提到的，各种 OOAD 方法在 20 世纪 90 年代中期是统一的。随后，开发了 UML。它被一些研究人员应用到了安全应用程序中，其中包括 Indrakshi Ray（见本章参考文献［RAY04］）的工作。随着 UML 的发展，它也被应用于面向切面的建模和分析。然而，随着服务技术的出现，UML 正在应用于服务建模中，我们期望这种方法将被应用于安全的服务。但是，我们必须注意，不要将服务人

为地建模为对象。因此，我们需要一种自下而上的方法对服务和安全的服务进行建模。

安全性已被纳入到软件工程的生命周期中，近期则是被应用到面向对象的生命周期中。例如，安全工程涉及定义安全策略，将安全性纳入系统设计、安全测试和维护。在面向对象的系统生命周期的情况下，安全性考虑将包括定义对象和活动的安全策略，以及将安全性纳入对象系统的设计、安全测试和维护。同样，在安全的面向服务的生命周期的情况下，我们需要确定安全策略、服务的安全级别以及服务之间的相互作用，包括服务的组成，以及将安全性纳入服务的设计和开发，之后测试安全服务。

正如我们已经讨论过的（见本章参考文献［THUR10］），托马斯·伊尔在他的书中关于 SOA 的部分，解释了服务的生命周期。他提出了 3 种发展服务的方式：一是自上而下的方法，二是自下而上的方法，三是他所谓的敏捷方法。在设计服务时，我们不应该在事后考虑安全性。我们必须考虑到自上而下、自下而上和敏捷方法的安全性。在自上而下的方法中，必须进行分析，然后设计服务，开发服务，测试服务，集成服务，最后维护服务。在这个方法中，安全策略必须贯穿整个过程。例如，当组合两个服务时，组合服务对应的策略是什么？在自下而上的方法中，服务根据需要进行设计和开发。因此，在设计服务时，必须考虑安全性。正如当设计新服务时，不能违反为更优先服务指定的安全策略。而在敏捷方法中，采用综合方法。也就是说，分析应用程序和确定服务同步进行，不必等待所有的服务被确定。对这种敏捷方法的安全性影响尚待研究。

考虑安全性时，需要注意另一个方面，即动态策略。也就是，在服务和服务组合上执行的安全策略可能随时间而变化。确保在适应不断变化的策略和安全级别时不会出现安全隐患是一项挑战。同时也是设计安全的面向服务系统的主要挑战。

下面，我们将讨论我们在 5.2 节中介绍的 SOAD 方法，并检查其安全性。第一步是分析应用程序并确定描述应用程序的服务，必须确定由每个服务封装的逻辑、逻辑的重用以及到该服务的接口。从安全角度来看，在定义服务时，我们必须考虑相应的一些安全策略。可能包括，该服务的安全级别是什么？在服务上执行的策略是什么？谁可以访问该服务？我们什么时候将服务分解成较小的服务，以防安全性不被侵犯？例如，服务 A 可能不能访问服务 B。但是，服务 B 可以被分解成服务 C 和 D，其中 A 可以访问 C 而不能访问 D。现在，如果 A 具有对 C 和 D 的访问，则可能会违反 A 不能访问 B 的策略。

下一步是服务之间的关系，其包括将要确定服务的组成。在自上而下的策略中，必须在进行服务的详细设计和开发之前确定所有服务和关系。对于大型应用设计，这可能是不可行的。在自下而上设计的策略中，必须确定服务才开始开发。在敏捷设计的策略中，将这两种策略融为一体。从安全的角度来看，可能存在着定义服务之间关系的策略。我们先前提供的关于服务 A、B、C 和 D 的示例显示，虽然 A 可以访问 C，但如果我们要执行 A 无法访问 B 的策略，则 A 可能无法访问 D。这种访问意味着调用特定服务。

在本章参考文献［THUR10］中，我们论述了业务逻辑可以被建模为服务。并且，这种方法为基于业务流程的 SOA 奠定了基础。业务流程基本上实现了工作流逻辑，使不同的应用程序能够相互交互。此外，我们已经讨论过，业务流程本身也可能被实现为

服务。因此，可以针对不同的应用程序调用业务流程服务，也可以实现为相互交互的服务。同时，业务服务也促进了再利用。从安全的角度来看，我们还要确定谁可以调用业务逻辑和业务流程服务。我们已经为工作流系统的安全性做了许多工作，其中包括 BFA（Bertino - Ferrari - Atluri）模型（见本章参考文献［BERT99］)。在这项工作中，我们需要审查关于业务逻辑和业务流程服务的原则。当服务被重复使用时，如果因此而产生互相冲突策略，将会发生什么？此外，我们还必须确保通过重用不会出现安全违规。

接下来，我们将考虑（见本章参考文献［THUR10］）中讨论的服务建模的关键点，并检查其安全性影响。其主要的问题是如何定义一个服务？在最高级别，诸如订单管理的整个应用程序可以是一项服务。但是，这是不可取的。在另一个极端情况，业务流程可以分为几个步骤，每个步骤都可以是一个服务。将执行一些特定任务的步骤分组到服务中是一项挑战。然而，当考虑到安全性时，我们不仅必须将执行某些特定任务的步骤分组到服务中，而且还必须确保这些分组能够有意义地执行。如果安全性是基于多层次的，那么我们可能需要为每个服务分配一个安全级别。以这种方式，服务可以由被准许的人在合适级别中执行。因此，将步骤分组是一项挑战，从任务角度和安全角度来看都是非常有意义的。

随后，我们必须检查候选服务并确定它们之间的关系。一个服务可能会调用其他服务，也可能组合两个服务来创建一个复合的服务。这意味着要确定边界和接口，并使组成和分离尽可能清晰。相互的依赖性可能导致更复杂的服务设计。服务操作可以是诸如执行计算的简单操作或诸如调用多个服务的复杂操作。再者，安全性可能会影响服务之间的关系。如果两个服务之间有一些关系，那么这两个服务都应该允许被在特定级别上的一组用户或多组用户使用。例如，如果服务 A 和 B 紧密集成，服务 C 不可能只可以访问 A 而不可以访问 B。如果 A 关于进行酒店预订，B 是关于出租车预约，则航空公司预订服务 C 应能够调用 A 和 B 两种服务。

一旦候选服务和服务操作被确定，下一步就是定义候选服务并描述服务的设计和服务操作。因此，从安全的角度来看，我们必须定义不仅有意义而且安全的服务和服务操作。而且，必须根据策略，执行候选服务到实际服务的映射。图 5.14 为安全的 SOAD 的基本流程。

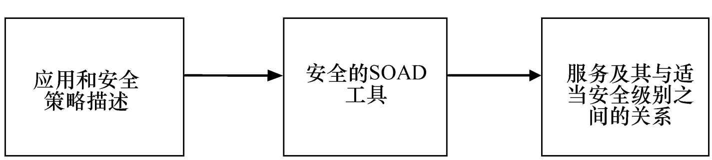


图 5.14　安全的 SOAD

正如我们在 5.2 节所论述的，SOAD 方法有许多种。接下来，我们将研究各种方法的安全性影响。如同神话般的 UML，我们相信能在统一的 SOAD 方法上取得进步。到那时，我们对这种方法的安全性会有一个更好的了解。

5.3.3.1　安全的 SOMA

如本章参考文献［SOMA］所述，SOMA 要实现 SOAD，需要通过身份识别、设计

规范和实现服务，实现可用于组合服务的服务组件和流程组件。要使用安全的SOMA，我们需要确定在服务和各种组件上执行的策略。对于多层次的安全WS，还需要分配服务的安全级别。此外，服务的执行级别也应该被定义。

5.3.3.2 安全的SOMF

如本章参考文献［THUR10］中所述，SOMF是面向服务开发生命周期的方法，并且支持许多建模实践和规则，有助于实现成功的面向服务的生命周期管理和建模。对此框架的安全性影响需要进一步检查。

5.3.3.3 用于服务的安全UML

用于服务的安全UML基本上形成了用于面向服务的分析和建模的安全UML。在安全应用中，应用UML和其他OOAD方法上已经做出的一些努力。我们需要进一步拓展这些方法来保护SOAD。还需要检查更多方面的安全性影响，例如面向服务的发现和分析建模、面向服务的业务集成建模、面向服务的逻辑设计建模、面向服务的概念架构建模和面向服务的逻辑架构建模等。

5.3.4 WS访问控制

我们在WS上的大部分工作都是基于访问控制。访问控制策略指要访问对象的主体必须满足的规则。我们已经为信息系统开发了几种访问控制策略，包括自主访问控制策略、强制访问控制策略，以及最近的基于角色的访问控制策略和使用控制策略。基于属性的访问控制（Attribute－Based Access Control，ABAC）正在被许多应用领域（包括国防部）所采用。作为一个模型，它是一种更适合诸如Web环境的开放型系统。此外，由OASIS等标准组织开发的一些模型，也是基于某种形式的属性访问控制模型。在本节中，我们将重点介绍访问控制的各种标准，然后讨论基于属性的访问控制。此外，我们还讨论了一些其他功能，例如建立Web环境中的信任，以及基于访问控制的推理控制方法。对于WS的访问控制的一些新兴标准分别是SAML和XACML。

5.3.4.1 SAML

本章参考文献［SAML］提出了单点授权，旨在“解决Web单点登录”问题。基于公钥/私钥基础设施，组中的一个身份提供者允许访问组内各项功能。在产品中提供SAML的公司包括：Microsoft Passport、OpenID（VeriSign）和全球登录系统（开源）。如SAML规范所述，其3个主要组成部分是：

声明：SAML有三种声明。认证声明是用户用于证明自己的身份（“约翰·史密斯在上午9点通过密码验证”）。属性声明包含有关用户的特定信息，例如他的支出限制（“John Smith销售经理每日旅行限支1000美元”）。授权决定声明确定用户可以做什么，例如他是否可以购买一个东西（“John Smith被允许购买指定的东西”）。

SAML权限：构成SAML声明的系统实体（也称为Identity Provider－IdP和Asserting Party）。

服务提供商：使用SAML声明的系统实体。

依赖方：使用接收到声明的系统实体（也称为SAML请求者）。

协议：定义了 SAML 要求和声明的方式。例如，现在使用的是 SOAP over HTTP，可能在将来会使用其他方法。

绑定：详细说明 SAML 消息交换如何映射到 SOAP 交换中。

身份管理的一个关键方面是 SAML，这就是身份信息如何从一个域传递给另一个域。SAML 2.0 版本将是 Liberty Alliance 建立其他的联合身份应用程序（如使用 Web 服务基于权限的属性共享）的基础。

SAML 配置文件是另一个重要的概念。它定义了核心协议和声明的约束和/或扩展，以支持特定应用程序对 SAML 的使用。它激活互操作性，并规定如何使用适当的协议消息通过指定的绑定来传达特定的语句。（例如，Web 浏览器 SSO Profile，指出如何通过许多不同的绑定使用身份验证查询和响应消息传递 SAML 身份验证声明，以便为浏览器用户启用 SSO）。通过允许支持特定的 SAML 配置文件（而不是完整的规范集），希望交换 SAML 消息的用户将更简单地获得互动操作。

SAML 的突出问题包括性能、联合和处理旧版应用程序。关于性能方面，其不支持缓存，并且必须通过 HTTP 使用 SOAP 来实现。此外，它不指定加密，造成的结果是可能会危及策略。对于联合，SAML 没有指定认证协议。此外，它无法处理多个域。因此，OASIS 正在审查联合身份管理。因为改进费用昂贵，所以 SAML 并不适用于旧版应用程序。

5.3.4.2　可扩展访问控制标记语言

XACML 是一种通用的授权策略模型和基于 XML 的规范语言。它独立于 SAML 规范，并具有三元组策略语法：<Object，Subject，Action>。它支持否定授权。XACML 策略处理器的输入/输出被明确地定义为 XACML 上下文数据结构。输入数据由 XACML－specifc 属性指示符以及 XPath 表达式引用。

策略由多个规则组成，较高级别的策略（PolicySet 元素）可以组合成一组策略。XACML 将多个规则组合到单个策略中。它允许多个用户拥有不同的角色，并将策略书写与应用环境进行分离。其目标是使访问控制语言规范化。一个策略有 4 个主要部分：目标、规则组合算法标识符、一组规则和约定。规则是策略的基本单位。规则的主要组成部分是：目标、效果、许可或拒绝、条件。实际上一个策略确定了一套资源、主题、操作和它所适用的环境。

XACML 包含以下一些元素。用户与资源进行交互。每个资源都被称为策略执行点（Policy Enforcement Point，PEP）的实体保护，它是语言实际使用，但不能实际上确定访问的地方。PEP 将其请求发送到策略决策点（Policy Decision Point，PDP）。策略实际可能会或可能不会存储在 PDP，但具有访问的最终决定权，同意或拒绝访问，然后再将决策转发给 PEP。在 XACML 的架构中，当客户端在服务器上发出资源请求时，PEP 负责执行访问控制策略。然而，为了执行这些策略，PEP 将在策略信息点（Policy Information Point，PIP）中形成描述请求者的属性，并将授权决定委托给 PDP。适用的策略位于由策略管理点（Policy Administration Point，PAP）管理的策略库中，并在 PDP 进行评估，然后 PDP 返回授权决定。基于此决定，PEP 可以向客户端提供适当的响应。如

5.3.4.2 节所述，XACML 请求是基于三元（主体、对象和动作）的。XACML 响应是以下之一：允许、合约许可、拒绝、不适用（PDP 无法定位目标与所需资源匹配的策略）、不确定（发生错误或某些必需值丢失）。

总之，XACML 协议的工作原理如下。策略管理点（PAP）创建安全策略并将这些策略存储在相应的存储库中。策略执行点（PEP）通过做出决策请求和执行授权决定来执行访问控制。策略信息点（PIP）作为属性值的来源、或策略评估所需的数据。策略决策点（PDP）评估适用的策略并做出授权决定。请注意，PEP 和 PDP 可能包含在同一应用程序中，或者可能分布在不同的服务器上。

XACML 的突出问题包括分布式响应和策略交叉引用。关于分布式响应，当 PEP 响应多个对象时会发生什么？当我们违背 PDP 或欺骗其通信时会发生什么？如何保证我们引用正确的对象？我们要承认这个事实：尽管系统是分布式的，但策略可能仍然只位于一个位置。在策略交叉引用方面，一项策略可能会涉及另一种策略。在继承或者与相关工作的联合/交互时，典型的问题就会出现。处理这种突发事件将是个挑战。图 5.15 为用于 WS 的 XACML 访问控制模型。

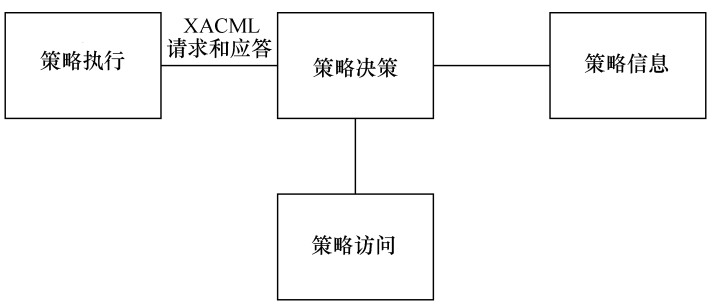


图 5.15　XACML 访问控制模型

基于属性的访问控制：XACML 基本上实现了基于属性的访问控制［ABAC］。虽然基于密码的访问控制在封闭环境中工作良好，但比如在网络这种开放的环境中，实现这样的机制是困难的。因此，基于属性的访问控制的概念是在 2000 年初形成的。通过这种方法，用户将会呈现他的证书。这些凭证将由某个证书机构颁发。如果需要，系统（或服务器）将使用多个证书机构验证用户的证书。一旦证书被验证通过，系统将检查证书的策略，并确定用户对资源的访问权。

ABAC 已经在多个系统中实现，包括国防部门的以网络为中心的企业服务和全球信息网格。ABAC 也可用于实现 RBAC（基于角色的访问控制，Role－Based Access Control）。在这种情况下，用户具有取决于自身角色的证书，基于此证书，用户被授予访问权限。从根本上来说，证书是用户的一些属性。最近，UCON（使用控制，Usage Control）模型已经被提出，这个模型是关于控制资源的使用以及控制对资源的访问的。以电话卡作为例子，当使用电话卡时，其余额被控制，并且访问取决于电话卡的余额，其实质上是剩余的使用时间量。将 ABAC 与 UCON 集成在一起是有可能的，有学者提出一个模型，就是基于

主体/用户的属性和资源的使用来控制访问（见本章参考文献［PARK04］）。

5.3.5 数字身份管理

身份管理（也称为联合身份管理或数字身份管理）与 WS 密切相关。在访问资源之前，用户以及 WS 必须进行身份验证。单点登录是一个流行的解决方案，一次登录为用户提供访问各种资源的服务。此外，SAML 目前为 WS 提供身份验证工具。然而，随着电子商务的监管要求，需要一个更强大的认证机制，这种机制被称为身份管理。

数字身份管理基础性的两个概念是 SSO 和联合身份管理。我们需要对相关概念以及为 SSO 和联合身份管理开发的技术和标准进行讨论。这包括自由联合体的工作、身份元系统及其信息卡实施、OpenID 项目和 Shibboleth。

如本章参考文献［SSO］中所述，SSO 是用户登录一次，可以在联合体中访问所有系统的性质。这样，用户只须登录一次，并且可以访问联合体或团体联合会或组织中的资源，而不会被提示需在每个组织中重新登录。目前存在两种类型的 SSO 机制：基于 Kerberos 和基于智能卡的。在 Kerberos 机制中，Kerberos 的票据授权票（Ticket - Granting，Ticket，TGT）用于授予凭证。在基于智能卡的登录中，用户使用智能卡进行登录。企业单点登录（Enterprise Single Sign - On，ESSO）提供了在访问多个应用程序时能够最小化密码和用户 ID 数量的支持。

如本章参考文献［FED］所述，“联合身份”或身份的“联合”描述了技术、标准和用例，以便在其他自治安全域中实现身份信息的可移植性。其中的用例包括跨域、基于 Web 的 SSO 等典型用例。目前，各种网站正在通过 Open ID 实施联合身份管理，其目标是确保一个域的用户以无缝的方式利用由另一个域所创建的所有技术。请注意，联合体是关于一起工作去执行任务（如 B2B 操作）或解决特定问题的组织。虽然这个想法已经存在了很多年，但最近随着 WS 的新兴标准出现，才使得我们现在可以拥有安全的联合体。在这样的联合体中，必须对用户访问资源进行管理，但不能对用户造成负担。

通过适当的联合身份管理，用户应能够跨域共享数据，支持 SSO 以及启用跨域用户属性管理。跨域 SSO 是联合身份管理的流行技术之一，然而最近已经有很多正在开发的新技术。联合身份管理标准的突出联合体之一是自由联盟，其他的工作就是开放式 ID 项目以及信息卡。本部分将概述身份管理的各种发展情况。

如本章参考文献［INFO］所述，身份元系统是一种“可互操作的数字身份架构，它使人们能够拥有并采用数字身份集合。这种数字身份集合需要基于多种底层技术、具体实施和提供者”。基于这种方法，用户可以继续维护他们的身份，并选择将为他们工作的身份系统，以便系统在迁移到不同的技术时还能够管理他们的身份。身份元系统的角色是身份提供者、依赖方和主体。身份提供者发行数字身份，依赖方的身份是需要各种服务的一方，主体包括最终用户和组织。

身份主要使用安全令牌的声明来表示。通过这些声明，身份提供者、依赖方和主体可以进行协商等操作。WS - Trust 和 WS - Federation 用于获取声明。双方之间的协商采

用 WS – Security Policy 和 WS – MetadataExchange 进行。用户对外产生的无缝操作由叫作 IdentitySelector 的客户端软件提供，该软件可以访问类似信息卡等相关技术。

信息卡是身份元系统的实现。如本章参考文献［INFO］所述，信息卡是个人数字身份，人们可以在线使用。信息卡可以是卡片形状的图片，人们可以使用这些卡来管理他们的身份。由于实现了身份元系统，所以参与信息卡实施的各方是身份提供者、依赖方和主体。身份选择器（如 Windows CardSpace）用于存储和管理用户身份。信息卡也支持 SSO，用户可以在一个地方登录并访问网络上的各种资源。

有两种类型的信息卡。一种类型是个人信息卡，使用户能够发布声明（例如姓名、电话等）并通知各个站点。另一种类型是管理型信息卡，身份提供者提出关于该用户的声明。

5.3.5.1 OpenID

如本章参考文献［OPEN］所述，OpenID 是一种开放的、分散的用户标识标准，允许用户以相同的数字身份登录许多服务。OpenID 本质上是一个 URL，用户被 OpenID 提供者认证。Symantec（赛门铁克公司）和 Microsoft 等许多公司都支持 OpenID。例如，Microsoft 提供 OpenID 与其 Windows CardSpace 之间的互操作性。OpenID 扩展了身份元系统的实体，并包含以下内容：

- 最终用户：想要向站点声明自己的身份的人。
- 标识符：最终用户选择的 URL 作为其 OpenID 标识符。
- 身份提供者或 OpenID 提供者：该实体提供注册 OpenID URL 的服务，并提供 OpenID 身份验证。
- 依赖方：要验证最终用户身份的网站（本质上是服务提供商）。
- 服务器或服务器代理：验证最终用户标识符的服务器。
- 用户代理：用户通过用户代理（例如浏览器）访问身份提供者或依赖方。

OpenID 的使用如下：首先，用户访问依赖方（例如，服务提供商）的网站来请求服务。这个依赖方有一个 OpenID 表单，它是用户的登录名。其次，用户将在逻辑过程之前给出自己的身份信息。从这些信息中，依赖方将发现身份提供者的网站。如本章参考文献［OPEN］所述，依赖方和身份提供者可能具有共享的密码，这个密码由关联句柄引用并被依赖方存储。然后，依赖方将用户的浏览器引导至身份提供者，以便用户可以与身份提供者进行身份验证，依赖方然后存储结果。依赖方将用户的 Web 浏览器重定向到身份提供者，以便用户可以与身份提供者进行身份验证。通常，身份提供者从用户那里请求密码，然后咨询用户是否要信任依赖方。如果用户拒绝此请求，则拒绝对服务的访问。如果没有，则用户浏览器被定向到具有用户凭证的依赖方。浏览器带着用户的证书被重定向到依赖方网站上的指定返回页面。这个过程中，认证方必须核实证书确实是来自身份提供者。

5.3.5.2 Shibboleth

Shibboleth 是一个分布式 Web 资源访问控制系统，允许联盟共同合作去共享基于 Web 的资源（见本章参考文献［SHIB］）。它定义了一个协议，携带身份验证信息和从

主站到资源站点的用户属性。然后，资源站点可以使用属性来对用户进行访问控制决策。这个基于 Web 的中间件层使用 SAML 实现。访问控制分阶段进行。在第一阶段，资源站点将用户重定向到其主站点，并获得由主站点认证用户的句柄。在第二阶段，资源站点返回主站点的属性权限的句柄，并返回一组用户的属性，依此进行访问控制决定。

Shibboleth 和 SSO 存在一些问题。资源网站如何知道用户的主站点？它又如何信任返回的句柄？答案是，它由系统信任模型处理。认证过程如下：当资源站点从用户请求主站点时，他从已经通过证书认证的受信任站点列表中选择它。句柄通过 SAML 签名以及消息进行验证。用户从列表中选择主站点。如果他已经注册，主站点会对用户进行身份验证。主站服务器身份验证后，会将具有 SAML 标记的消息返回给目标资源站点。资源站点（如果标记匹配）然后为用户提供别名（句柄），并向主页发送声明消息，以确定用户必需的属性是否可用。为了确保隐私，系统每次为用户的身份提供不同的别名。它每次都需要已发布的用户属性策略，从而实现对目标站点中权限属性的控制。一般情况下，协议属性发布策略处于用户和管理员之间。

信任是 Shibboleth 的核心。它完全信任在联盟中注册的目标资源站点和原始主站点。现有的信任模型的缺点是认证机构和属性权限之间没有区别。我们需要有一个允许更复杂的信任分配范围，例如静态或动态认证授权。现有信任模式的另一个缺点是它只提供基本的访问控制功能。它缺乏许多应用程序所需的灵活性和复杂性，它们必须根据角色层次结构或各种约束（如每天的时间或职责分离）来提供访问控制决策。

在基本的 Shibboleth 中，目标站点信任原始站点，以对其用户进行身份验证并正确管理其属性，而原始站点信任目标站点，以向其用户提供服务。信任是由数字签名的 SAML 消息来传达的，签名时使用了目标和源服务器密钥对。每个 Shibboleth 系统的每个站点中只有一个密钥对。因此，每个 Shibboleth 系统只有一个信任点。所以这需要一个更细粒度的分布式信任模型，并且能够使用多个原始权限来发布和签署身份验证和属性声明。多个机构应该能够向用户发布属性，目标站点也能够验证发行者与用户之间的绑定。这样做的目标是能够在其策略中说明，发布哪一个其信任的属性权限，哪些属性属于哪些用户组。在进行访问控制决策时，目标站点能够独立于发布站点的属性和权限信任。但是，并非所有属性发布授权机构都是原始站点的一部分。如果目标站点具有由多个授权发布的属性，则应允许用户访问其资源。信任基础架构应支持动态授权，以便特权属性的持有者可以将其（一小部分）委派给别人，而不必重新生成系统中的任何内容。目标站点应该能够决定它是否真正地信任原始的属性存储库。如果不是，则可以要求更强大的属性授权权限，而不是由发送 Web 服务器的 SAML 签名授予的权限。

Shibboleth 定义了各种信任模型，这些模型已经使用 X. 509 实现。我们可以从两个不同的方面来看待信任：

一是在属性发布授权机构的信任分配。

二是原始站点属性存储库的可信赖性。

信任模型及其实现的进一步细节，以及授权和隐私问题在本章参考文献［TRUST］

中有讨论。

5.3.5.3 自由联盟

自由联盟的成立是为了促进身份管理的标准。它现在包括 100 多个成员，其中包括技术开发商和提供商以及消费者。自由身份联合会（也称为身份联盟）和自由身份 Web 服务（也称为身份 WS）是其发布的两个主要服务功能。

身份联合使得网络用户（例如，电子商务用户）可以进行身份验证和登录到域，从而可以访问多个服务。同时，它也是 SAML 2.0 的基础。如本章参考文献［LIB］所述，身份 WS 标准是部署和管理基于身份的 WS 的开放框架。这些 WS 应用程序包括地理位置、联系人通信簿、日历、移动消息和自由人员服务。通过这些服务，可以以隐私保护的方式在网络上管理书签、博客、照片共享和相关社会服务。隐私和策略管理是自由联盟工作的关键方面。在本章参考文献［LIB］中也指出，全球已经有超过十亿个支持自由的设备被跟踪。身份联合最新的功能包括身份管理框架和身份保证框架。身份管理框架是支持身份存储和管理的标准集合，它使用 LDAP（轻量级目录访问协议，Lightweight Directory Access Protocol）、SAML 和 WS – Trust 标准。身份保证框架支持四个身份保证级别，这些级别由国家标准与技术研究所确定。图 5.16 说明了各种身份管理技术。

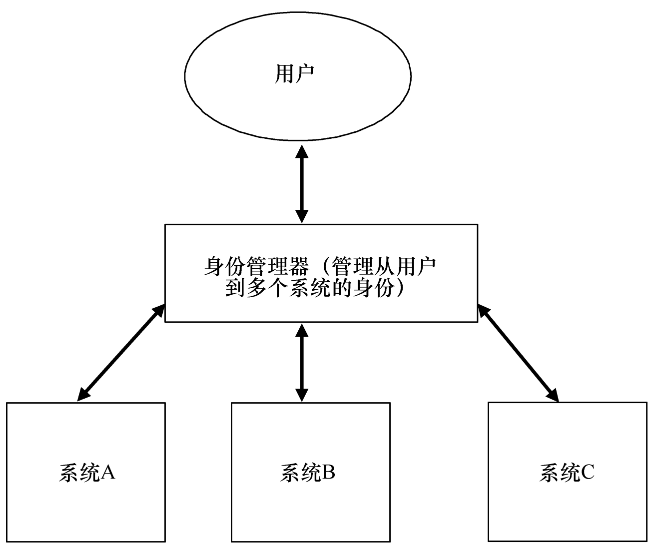


图 5.16　身份管理

5.3.6 WS 安全模型

WS 安全中的大部分工作都侧重于访问控制模型。也就是说，访问控制策略将确定用户能否对 WS 提供资源的访问。基于访问控制模型已经开发了几种类似 XACML 的标准。然而，对于许多应用程序，访问控制模型并不能满足其需要。例如，在复合 WS 的情况下，一个 Web 服务 S1 可以调用另一个 Web 服务 S2。在这样的调用中，S1 的权限将被执行，而不是调用 S1 的用户 U 的权限。这意味着返回给 U 的信息可能是用户未被

授权获取的信息。为了避免这样的安全性损害，用户 U 必须将其权限委派给 S1，以便在 S1 调用 S2 时使用 U 的权限。这种调用由所使用的授权模型来管理。

复合 WS 的另一个安全问题是信息流。也就是说，当 WS 组成时，关键是不能存在从高级别到低级别的信息流。我们的研究重点是 WS 安全性的各个方面，包括 Web 服务组合的授权模型和信息流。因此，在本节中，将概述我们在 WS 安全性方面的研究。同时，特别总结了我们最近的论文（见本章参考文献［SHE07］、［SHE08］和［SHE09］）中的报告。在本节中，还将介绍 WS 的授权（委托）模型以及服务组合中的信息流，并讨论 WS 的多级安全性。图 5.17 说明了 WS 的安全模型。

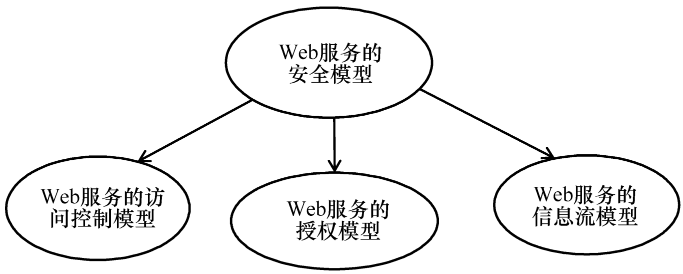


图 5.17　Web 服务的安全模型

5.3.6.1　授权模型

访问控制模型指定主体对客体的访问权限。但是，它没有指定调用 WS 的策略，这需要我们有适当的策略来调用 WS。例如，假设服务 S1 调用服务 S2 且 S1 无法访问资源 X，而 S2 可以访问资源 X，如果 S2 必须访问 X 并将 X 返回到 S1，则会出现安全冲突。这意味着当 S2 代表 S1 访问 X 时，则 S1 的权限必须传递给 S2。在上述示例中，这样的策略将起作用，因为 S2 具有 S1 不具备的其他凭证。问题是如果 S2 无法访问 X，而 S1 访问 X，会发生什么。如果 S1 的凭证传递给 S2，则 S2 将可以访问 X。

我们对 WS（见本章参考文献［SHE07］、［SHE08］）的授权模型进行了广泛的研究。我们认为，授权模型必须是弹性的。在某些情况下，在不会违反访问控制策略的情况下，即使只是有限的凭证，S2 也可以操作。也就是说，如果 S1 调用 S2，则 S2 将不会访问 S1 无法访问的任何资源。然而，在一些情况下，S2 可能需要使用其全部凭证进行操作，那个时候，S2 必须决定哪些信息可以传递给 S1。

如果没有重叠的凭证，授权模型变得更加复杂。也就是说，如果 S1 和 S2 没有任何共同的凭证，则 S1 无法调用 S2。例如，如果 S1 以教授的身份运行，而 S2 则以秘书的身份进行操作。如果教授和秘书没有任何共同的凭证，则教授不能要求秘书执行某些功能。在这种情况下，系统必须确定如何委托。责任分工是安全模型的重要条件。这里的挑战是如何将责任的分工情况转换到 WS 的安全模型中。为了 WS 的组合和基于链的 WS，将访问控制模型与委托模型进行结合需要大量研究工作。基于链的 Web 服务的形式为 S1 调用 S2，S2 调用 S3 和 S3 调用 S4。图 5.18 说明了授权模型。

5.3.6.2　信息流模型

要了解信息流模型，我们需要回顾历史上的模型。早在 1973 年，Bell 和 LaPadula

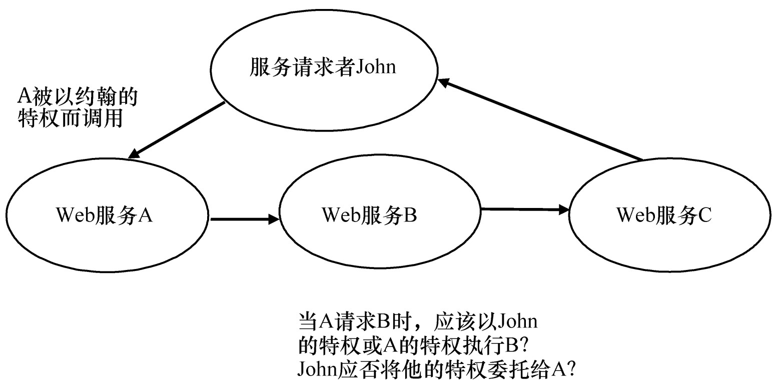


图 5.18　授权模型

模型被开发用于访问控制。虽然这种模型阻止了一个低级主体直接获取高级数据的行为，但并不能防止非法信息流的流动。例如，通过操纵文件中的锁，数据可以被隐蔽地从高级别主体传递到低级别主体。为了防止这种流动，Goguen 和 Meseguer 在 1982 年前后开发了非干扰模型。使用这种模型，数据不可能从高级别主体流向低级别主体。本质上，高级别主体的行为并不妨碍低级别主体。我们对 WS 的研究应用了类似的原理。

到目前为止，已经做了很多工作去防止 WS 中出现非法信息流。然而，以前的工作侧重于以下两个方面。每个 Web 服务均满足安全属性，WS 的组合也满足了安全属性。但是，中介服务之间的信息流未被考虑。例如，如果 WS S1 由 WS S2、S3 和 S4 组成，而最终结果是安全的，则从 S3 到 S1 仍然可能存在非法的信息流。图 5.19 说明了信息流模型。

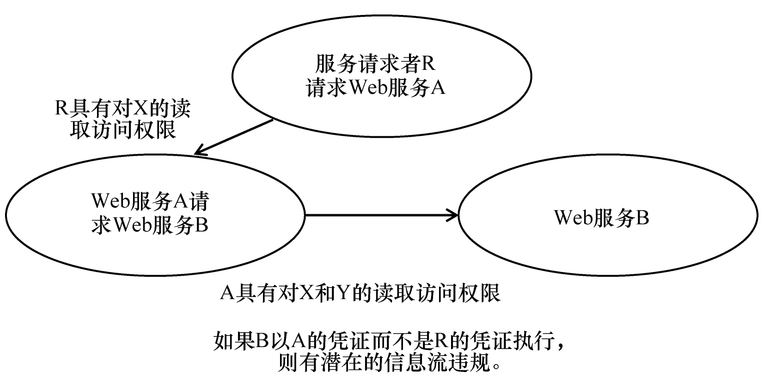


图 5.19　信息流模型

我们要寻求新的方法去阻止这种信息流。以前的模型做出的另一个假设是 WS 的组成是由一个可靠的过程执行的。当在 Web 环境中存在多个安全域时，这是不现实的。因此，我们不能做这样的假设。我们的工作也没有这样的假设。

我们在 WS 信息流模型方面的工作详见本章参考文献［SHE09］。特别是，我们定

义了转换因子，它用来测量从输出推测的输入和服务的逻辑数据之间的相似性。反过来又用来确定信息是否从较高级别的服务非法流向较低级别的服务。我们还开发了协议，使组合过程不被信任。然后，我们开发了用于在 Web 环境中协作执行安全验证的算法。

5.3.6.3　多级安全 WS

WS 的大部分安全研究都是基于自主访问控制。特别地，基于属性的访问控制正在被应用到面向 WS 的许多模型中。我们的研究重点是授权模型和信息流模型，这些模型为了追求无干扰，受到了 Goguen 和 Meseguer 模型的影响。

WS 的多层次安全性还没有太多的工作报告。不过，我们认为多层次安全是一个重要的方面。我们需要确保 Bell 和 LaPadula 模型的简单安全性和 * 属性不仅能得到满足，而且 Gouge 和 Meseguer 的无干扰属性也可得到满足。在 Bell 和 LaPadula 模型中，如果主体的安全级别高于客体的安全级别，则主体可以从对象中读取信息。如果主体的安全级别低于客体的安全级别，则主体向客体写入信息。另外，对于 WS 我们还需要调用策略。也就是说，如果 S1 服务描述的安全级别高于 S2 服务描述的安全级别，则服务 S1 调用另一服务 S2。然而，当 S1 调用 S2 时，S2 将在 S1 的安全级别进行工作。这种方式 S2 将拥有 S1 的全部凭证。因此，如果 S2 服务描述安全级别高于 S1，则 S1 不能调用 S2。例如，如果 S2 的服务描述是隐秘的，并且 S1 的服务描述未被分类，且 S1 在未分类级别操作，则 S1 不能调用 S2。假设，如果允许 S1 调用 S2，则 S2 必须以未分类的级别操作，这是一个问题，因为 S2 的描述是隐秘的。现在假设 S1 和 S2 的服务描述满足了这些策略。然而，如果 S1 在未分类操作，并且 S2 被允许在秘密级别操作，则 S2 不能将任何结果发送回 S1，因为它将违反 * 属性。

5.4　总结和展望

本章概述了面向服务的计算相关概念和服务的安全问题。正如我们所说，服务是云计算的核心和灵魂。这是因为云计算作为服务提供给客户。首先，我们讨论了服务概念、SOA 和 WS、新兴的 X 服务模式和 SOAD。然后，又讨论了 SOA 和 WS 的安全性。特别地，专门讨论了 WS 的访问控制，比如 SAML 和 XACML 的标准，以及一些新兴的安全模型，如授权模型、信息流和多层次安全性。另外，还阐述了 WS 的身份管理。

有几个领域需要将来的研究和开发。虽然 WS 有许多发展，但是面临的主要挑战是语义 Web 技术的应用和 WS 的安全性保护。此外，全球信息网格和以网络为中心的企业服务等重大设施都是基于 WS 和 SOA。因此，通过使用语义网络来确保这些技术得以实现，以及使 WS 更加智能化对于下一代网络来说至关重要。

对于 SOAD，我们没有标准的方式来建模和分析服务以及安全的服务。目前，我们还需要一个合适的服务安全模型。ABAC 就是一个这样的模型。我们需要检查 ABAC 如何与 UCON 集成。还需要更详细地检查服务的推理问题。最后，我们还要开发类似于 SAML 和 XACML 的标准，以包含更复杂的细粒度访问控制形式。随着 WS 的广泛应用，我们在网络上进行越来越多的事务，例如我们参与社交网络。因此保护个人身份以及确

保授权访问至关重要。此外，用户可能涉及多个社交网络和多个事务。用户可能在不同的系统中具有不同的身份。所以我们需要有效的机制来管理可能数十亿用户的众多身份。身份管理研究刚刚开始。我们还需要在这方面做更多的工作，包括制定适当的标准。最后，关于安全模型，虽然大部分工作都集中在 WS 的访问控制模型上，但我们已经对授权模型和信息流模型进行了一些工作。使用授权模型，它的目标是使服务将其凭证授权给其调用执行的另一个服务。使用信息流模型，目标是确保在服务组合期间信息不会从高级别传递到低级别。虽然 WS 的访问控制模型相当先进，但授权模型和信息流模型的研究还处于起步阶段。我们的工作只探讨了一些初步的想法，还需要正式指定具体的安全属性，并证明服务对于授权模型和信息流模型是安全的。另外，还需要检查访问控制、授权和信息流模型的集成。

作为参考，我们已经给出了各种 Web 服务和安全 WS 技术、标准和协议的 URL。应该注意的是，这些 URL 可能会改变，因此我们督促读者关注 W3C 和 OASIS 网页以及微软和 IBM 等企业开发标准的网页，以跟上技术的发展。如 5. 1 节所述，云计算通常作为服务集合提供给消费者。因此，本章讨论的主题是构成了云计算和云计算安全的基本内容。

参 考 文 献

[ABAC] Information System Security Operation: Attribute-Based Access Control, http://www.docstoc.com/docs/44587189/Information-System-Security-Operation-Attribute-Based-Access-Control

[ADDRESS] Web Services Addressing, http://www.w3.org/Submission/ws-addressing/

[AMAZON] Amazon.com, http://aws.amazon.com/

[APAC] The Apache Software Foundation Web Services Project, http://ws.apache.org/

[ASAP] OASIS Asynchronous Service Access Protocol, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=asap

[BERT99] Bertino, E., E. Ferrari, and V. Atluri. The specification and enforcement of authorization constraints in workflow management systems. *ACM Transactions on Information Systems Security (TISSEC)*, 2(1): 65–104, 1999.

[BERT06] Bertino, E. and L. Martino. Security in SOA and web services. *IEEE International Conference on Services Computing (SCC 2006)*. Chicago, IL, USA.

[BPEL] http://en.wikipedia.org/wiki/Business_Process_Execution_Language

[CAF] Web Services Composite Application Framework, http://en.wikipedia.org/wiki/WS-CAF

[CONV] WS-Secure conversation, http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html

[COORD] Web services transactions specifications, http://www.ibm.com/developerworks/library/specification/ws-tx/#coor

[ENUM] WS-Enumeration, http://www.w3.org/Submission/2006/02/

[ERL05] Erl, T. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*, Prentice-Hall, New Jersey, 2005.

[EVENT] WS-Eventing, http://www.w3.org/Submission/WS-Eventing/

[FED] Federated Identity, http://en.wikipedia.org/wiki/Federated_identity_management

[GUTI10] Gutierrez, C., E. Fernandez-Medina, and M. Piattini (Editors), *Web Services Security Development and Architecture*, Information Science Reference, Hershey, PA, 2010.
[IBM] Transformation to SOA, http://www.ibm.com/developerworks/rational/library/08/0115_gorelik/
[INFO] Information card, http://en.wikipedia.org/wiki/Information_Card
[LIB] Liberty Alliance, http://en.wikipedia.org/wiki/Liberty_Alliance
[MEP] Message Exchange Pattern, http://en.wikipedia.org/wiki/Message_Exchange_Pattern
[META] Web Services Metadata Exchange, http://www.ibm.com/developerworks/library/specification/ws-mex/
[MODEL] Analysis and design techniques for service-oriented development and integration, http://www.perspectivesonwebservices.de/download/INF05.ServiceModelingv11.pdf
[NOTIFY] WS-Notification, http://www.ibm.com/developerworks/library/specification/ws-notification/
[OASIS] OASIS Standards, http://www.oasis-open.org/specs/
[OMG] Object Management Group, http://www.omg.org
[OPEN] OpenID, http://en.wikipedia.org/wiki/OpenID
[PA] WS-Policy assertions, http://xml.coverpages.org/ws-policyassertionsV11.pdf
[PARK04] Park, J. and R. S. Sandhu, The $UCON_{ABC}$ usage control model. *ACM Transactions on Information Systems Security*, 7(1): 128–174, 2004.
[POLICY] WS-Policy, http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/
[RAY04] Ray, I., R. B. France, and N. GeriGeorg, An aspect-based approach to modeling access control concerns. *Information & Software Technology*, 46(9): 575–587, 2004.
[RELIABILITY] WS-Reliability, http://www.service-architecture.com/web-services/articles/web_services_reliability_ws-reliability.html
[REST] Representational State Transfer, http://en.wikipedia.org/wiki/Representational_state_transfer
[REST-SOAP] REST vs. SOAP at Amazon, http://www.oreillynet.com/pub/wlg/3005
[SAML] Security Assertion Markup Language, http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language.
[SELL93] Sell, P. J. and B. M. Thuraisingham, Applying OMT for designing multilevel database applications. *Database Security, VII: Status and Prospects, Proceedings of the IFIP WG11.3 Working Conference on Database Security*, Lake Guntersville, Alabama, USA, 12–15 September, 1993, pp. 41–64.
[SHE07] She, W., B. M. Thuraisingham, and I.-L. Yen, Delegation-based security model for web services. *IEEE International Symposium on High Assurance Systems Engineering (HASE 2007)*, November 2007, Dallas, TX, USA, pp. 85–91.
[SHE08] She, W., I-L. Yen, and B.M. Thuraisingham, Enhancing security modeling for web services using delegation and pass-on. *2008 IEEE International Conference on Web Services (ICWS 2008)*, September 2008, Beijing, China, pp. 545–552.
[SHE09] She, W., I-L. Yen, B. M. Thuraisingham, and E. Bertino, The SCIFC model for information flow control in web service composition. *IEEE International Conference on Web Services, ICWS 2009*, Los Angeles, CA, USA, July 2009, pp. 1–8.
[SHIB] Shibboleth Internet 2 http://en.wikipedia.org/wiki/Shibboleth_%28Internet2%29
[SOAD] Service Oriented Analysis and Modeling, http://en.wikipedia.org/wiki/Service-oriented_modeling
[SOMA] Sevice Oriented Modeling and Analysis, http://www.ibm.com/developerworks/library/ws-soa-design1/

[SRIV] Srivastava, B. and J. Koehler, *Web Service Composition—Current Solutions and Open Problems*, http://www.zurich.ibm.com/pdf/ebizz/icaps-ws.pdf
[SSO] Single Sign On http://en.wikipedia.org/wiki/Single_sign-on
[SURVEY] A Survey of Service Oriented Development Methodologies, http://www.dcs.shef.ac.uk/~ajhs/research/papers/soasurvey.pdf
[THUR94a] Thuraisingham, B. and A. Schafer, Applying OMT for real-time applications, *Proceedings of the IEEE Real-time Applications Workshop*, Bethesda, MD, July 1994.
[THUR94b] Thuraisingham, B. Applying OMT for healthcare applications, *Proceedings IEEE Dual Use Technology Conference*, Rome, NY, May 1994.
[THUR10] Thuraisingham, B. *Secure Semantic Service Oriented Systems*, CRC Press, Boca Raton, FL, 2010.
[TRANS] Web Services Transactions Specifications, http://www.ibm.com/developerworks/library/specification/ws-tx/
[TRUST] WS-Trust, http://en.wikipedia.org/wiki/WS-Trust
[UML] Unified Modeling Language, http://en.wikipedia.org/wiki/Unified_Modeling_ Language
[WSIL] Web Services Inspection Language, http://www.ibm.com/developerworks/library/specification/ws-wsilspec/
[WSS] WS-Security, http://en.wikipedia.org/wiki/WS-Security
[XACML] XACML, http://en.wikipedia.org/wiki/XACML
[ZHAN07] Zhang, L.-J., J. Zhang, and H. Cai, *Services Computing*, Springer, Heidelberg, 2007.

第 6 章　语义 Web 服务和安全

6.1　概述

在前一章中，我们讨论了服务计算和安全服务计算。虽然服务正在成为云计算的一个重要方面，但目前的服务并不能使用语义。此外，虽然目前的网络技术使得从语法角度整合信息变得便利，但是仍然需要做很多工作来处理各种系统和应用程序的不同语义。也就是说，目前的网络技术在信息管理和集成方面很大程度上取决于“人的参与”。在本章中，我们将讨论语义 Web 技术以及 WS 如何利用这些技术，使其具有语义功能。在这里，如果一个云利用了语义 Web 技术，我们就定义为语义云。稍后在本书中，我们讨论基于语义云的几个实验系统。

WWW 之父 Tim Berners Lee 意识到当前网络技术的不足之处，并努力使网络更加智能化。他的目标是建立一个网络，根本上缓解人类必须整合不同信息来源，并进行广泛搜索的负担。他得出的结论是，需要机器可理解的网页，和使用本体进行信息集成。这导致了语义 Web（见本章参考文献［LEE01］）的概念的产生。利用语义 Web 技术的 WS 是语义 Web 服务。

语义网可以被认为是一个高度智能和复杂的网络，因此很少或根本不需要人力干预，来执行任务，比如调度任命、协调活动、搜索复杂的文档以及整合不同的数据库和信息系统等。虽然在开发这样一个智能网络方面取得了很大进展，但仍有很多工作要做。例如，本体匹配、智能代理和标记语言等技术正在为开发语义 Web 做出了很大的贡献。但目前的语义 Web 仍然需要人类做出决定并采取行动。

最近在语义 Web 上取得了很多进展。W3C 规定了语义 Web（见本章参考文献［W3C］）的标准。这些标准包括 XML、RDF 和互操作性的规范。当然，语义 Web 的安全也是非常重要的。也就是说，构成语义网的组件必须是安全的，这些组件包括 XML、RDF 和本体。另外，我们需要安全的信息集成，还需要检查语义 Web 的信任问题。因此，一个重要的工作就是开发安全的语义 WEB 标准，包括安全 XML、安全的 RDF 和安全的互操作性规范（请参见本章参考文献［THUR05］）。在本章中，我们将讨论语义 Web 的各种组件和语义 Web 服务。

虽然代理对管理数据和语义 Web 上的活动至关重要，但一般而言，有些人不将代理视为语义 Web 技术的一部分，而另一些则将代理视为语义 Web 的一部分。由于代理的主体很广泛，也花了许多努力在开发代理和安全代理上，所以我们不将代理作为本书的一部分进行论述。但是我们还会在本书中提到代理，因为这些代理能使用 XML 和 RDF，并且理解数据和 Web 页面。代理代表用户行事，代理能使用明确的协议相互通

信。根据执行任务的不同，可以开发出各种类型的代理，包括移动代理、智能代理、搜索代理和知识管理代理。代理调用 WS 来执行操作。有关代理的详细信息，请参阅本章参考文献［HEND01］。

随着对数据和信息管理需求的增加，还需要维护数据库、应用程序和信息系统的安全性。必须保护数据和信息，免受未经授权的访问以及恶意的损坏。随着网络的普及，保护数据和信息变得更加重要，因为现在许多人可以访问这些数据和信息。因此，我们需要有效的机制来保护语义 Web 技术。特别是，我们需要保护 XML 和 RDF 文档以及其他组件，如安全本体和安全 Web 规则。

本章的内容如下。在 6.2 节中，我们将提供语义 Web 技术的概述。其中我们将讨论由 Tim Berners Lee 所指定的语义 Web 的分层架构，如 XML、RDF、本体以及 Web 规则和语义 Web 服务等。6.3 节讨论语义 Web 技术的安全性。本章总结在 6.4 节。图 6.1 说明了本章讨论的相关概念。大量关于语义 Web 的讨论是从 Antoniou和 van Harmelen 所著的一书中（见参考文献［ANTO08］）总结过来的。关于最新的规范，请参考本章参考文献［W3C］。

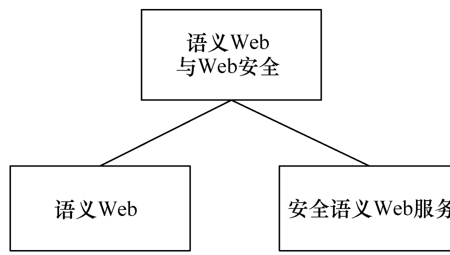


图 6.1　语义 Web 服务与安全

6.2　语义 Web

6.2.1　分层技术栈

图 6.2 为语义 Web 技术栈，这是 Tim Berners Lee 开发的架构。本质上，语义 Web 由各层构成，每一层均利用了下一层的技术。最低层是协议层，它通常不包括在语义技术的讨论中。上一层是 XML 层。XML 是文档表示语言，将在 6.2.2 节中讨论。虽然 XML 规定语法是足够的，但是比如“文档 D 的创建者是 John”这样的语义很难在 XML 中规范说明。因此，W3C 开发了使用 XML 语法的 RDF，我们在 6.2.3 节描述 RDF。



图 6.2　语义 Web 技术栈

然后，语义 Web 社区进一步出现，并引出了基于语言（例如 OWL）的本体规范。需要说明的是，OWL 解决了 RDF 的不足之处。我们在 6.2.4 节讨论 OWL。为了制定出各种策略，语义 Web 社区提出了一种 Web 规则语言，如语义 Web 规则语言（Semantic Web Rules Language，SWRL）和规则标记语言（Rules Markup Language，Rules ML）。6.2.5 节中讨论规则，第 6.2.6 节讨论语义 Web 服务。

6.2.2 XML

由于 HTML 的限制和标准通用标记语言（Standard Generalized Markup Language，SGML）的复杂性，XML 显得更加需要。它是由 W3C 指定的可扩展标记语言，旨在使 Internet 上的结构化文档交换更加容易。XML 的一个重要方面是用于文档类型定义（Document Type Definition，DTD），从而定义正式模型中每个元素的作用。XML 模式现在已经成为规范体系结构的关键。XML 模式也是 XML 文档。本节将讨论 XML 的各种组件，包括：语句、元素、属性和架构。XML 的组件如图 6.3 所示。

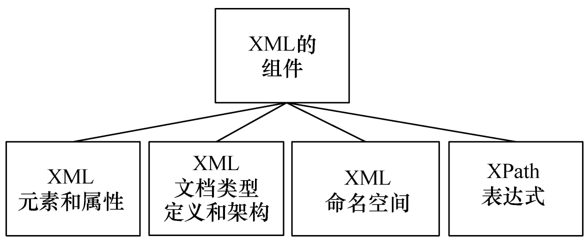


图 6.3　XML 的组件

6.2.2.1　XML 语句和元素

以下是一个 XML 语句的例子，它描述了一个事实“John Smith 是得克萨斯州（Texas）的一名教授”，元素是 name 和 state。XML 语句如下：

```
<Professor>
       <name> John Smith </name>
       <state> Texas </state>
</Professor>
```

6.2.2.2　XML 属性

假设我们要说明有一个教授，名叫 John Smith，他的薪资 6 万美元。我们可以使用元素或属性来说明。下面的示例显示了属性 Name 和 Salary 的使用。

```
<Professor
       Name = "John Smith", Access = All, Read
       Salary = "60K"
       </Professor>
```

6.2.2.3　XML DTD

DTD 本质上说明了 XML 文档的结构。对于具有 Name 和 State 元素的教授，请考虑以下 DTD。这将被编码为

```
<!ELEMENT Professor Officer (Name, State)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT access (#PCDATA).>
```

6.2.2.4　XML 模式

虽然 DTD 早期尝试为 XML 文档制定结构，但是 XML 模式本身更加适合用于说明结构。与 DTD 不同，XML 模式基本上使用 XML 语法进行说明。请考虑以下示例：

```
<ComplexType = name = "ProfessorType">
       <Sequence>
       <element name = "name" type = "string"/>
       <element name = "state" type = "string"/>
       <Sequence>
</ComplexType>
```

6.2.2.5 XML 命名空间

命名空间用于消除歧义。下面给出了一个例子。

```
<CountryX: Academic-Institution
       Xmlns: CountryX = http://www.CountryX.edu/Institution DTD"
       Xmlns: USA = "http://www.USA.edu/Institution DTD"
       Xmlns: UK = "http://www.UK.edu/Institution DTD"
<USA: Title = College
       USA: Name = "University of Texas at Dallas"
       USA: State = "Texas"
<UK: Title = University
       UK: Name = "Cambridge University"
       UK: State = Cambs
</CountryX: Academic-Institution>
```

6.2.2.6 XML 联合/分开

XML 数据可以被分开，并且数据库可以形成联合。这将在下面的段落中说明。

Site 1 document：

```
<Professor-name>
       <ID> 111 </ID>
       <Name> John Smith </name>
       <State> Texas </state>
</Professor-name>
```

Site 2 document：

```
<Professor-salary>
       <ID> 111 </ID>
       <salary> 60K </salary>
</Professor-salary>
```

6.2.2.7 XML – QL、XQuery、XPath、XSLT

XML – QL 和 XQuery 是针对 XML 提出的查询语言，XPath 用于指定查询。基本上，可以使用 XPath 表达式来表示 XML 语句中的特定元素。在我们的研究中，将策略规则指定为 XPath 表达式（参见本章参考文献［BERT04］）。XSLT 用于呈现 XML 文档。其细节在本章参考文献［W3C］以及［ANTO08］中给出。另一个有用的参考文献是本章参考文献［LAUR00］。

6.2.3 RDF

虽然 XML 是指定各种语句中语法的理想选择，但是很难使用 XML 指定语句的语义。例如，使用 XML，很难指定如下语句：

■ 工程师是员工的子类；

■ 工程师继承员工的所有属性。

请注意，上述语句指定类/子类和继承关系。RDF 由 Tim Berners Lee 及其团队开发，使得 XML 的不足之处得以解决。RDF 使用 XML 语法，也需要额外的构件，我们对其中一些结构进行讨论。详细信息请参见本章参考文献［ANTO08］。

RDF 是语义 Web 的本质。它基于 XML 语法，使用本体对各种语句提供语义。RDF 概念包括由资源、属性和语句组成的基本模型，以及由单元组、序列、待选内容组成的容器模型。我们将讨论一些基本概念。RDF 的组件如图 6.4 所示。

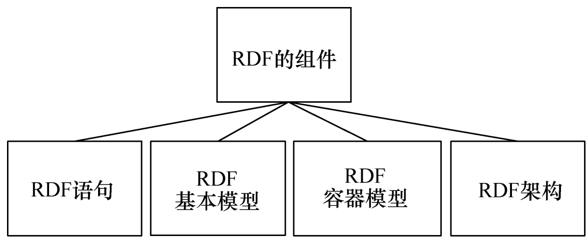


图 6.4　RDF 的组件

6.2.3.1　RDF 基础

RDF 基本模型由资源、属性和语句组成。在 RDF 中，一切就像人、车、动物等资源。属性描述了资源之间的关系，像"买""发明"和"吃"等。语句是一个三元组形式：(对象，属性，值)。语句例子如下：

■ Berners Lee 发明了语义 Web；

■ Tom 吃了苹果；

■ Mary 带来了一件衣服。

图 6.5 说明了 RDF 中的一个语句。在这里，Berners Lee 是对象，语义 Web 是值，而发明是属性。

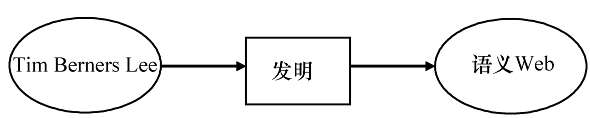


图 6.5　RDF 语句

6.2.3.2　RDF 容器模型

RDF 容器模型由单元组、序列和待选内容组成。如本章参考文献［ANTO08］中所述，这些结构在 RDF 中指定如下：

Bag：无序的容器，可能包含多个事件

- Rdf：Bag

Seq：有序容器，可能包含多个事件

- Rdf：Seq

Alt：一组待选内容

- Rdf：Alt

6.2.3.3　RDF 规范

如本章参考文献［ANTO08］所述，RDF 规范已经给出了属性、类型、嵌套、容器等。例如：Berners Lee 是《语义 Web》一书的作者。

上述语句说明如下（另见本章参考文献［ANTO08］）：

```
<rdf: RDF
  xmlns: rdf = "http://w3c.org/1999/02-22-rdf-syntax-ns#"
  xmlns: xsd = "http://- - -
  xmlns: uni = "http://- - - -
<rdf: Description: rdf: about = "949352"
  <uni: name = Berners Lee </uni:name>
  <uni: title> Professor <uni:title>
</rdf: Description>
<rdf: Description rdf: about: "ZZZ"
  <uni: bookname> semantic web <uni:bookname>
  <uni: authoredby:Berners Lee <uni:authoredby>
</rdf: Description>
</rdf: RDF>
```

6.2.3.4 RDF 模式

XML 模式指定了 XML 文档的结构，可以被认为是元数据，而 RDF 模式指定了类/子类之间的关系。例如，我们需要 RDF 模式来指定诸如工程师是员工的子类等语句。以下是此语句的 RDF 规范。

```
<rdfs: Class rdf: ID = "engineer"
<rdfs: comment>
The class of Engineers
All engineers are employees
<rdfs: comment>
<rdfs: subClassof rdf: resource = "employee"/>
<rdfs: Class>
```

6.2.3.5 RDF 公理语义学

第一阶逻辑用于指定公式和推论。需要用到以下构件：

内置函数（第一个）和谓词（类型）

演绎推理：从 A 和如果 A 那么 B，推导 B

以下示例取自本章参考文献［ANTO08］：

例子：

所有容器都是资源，那就是如果 X 是一个容器，那么 X 就是一个资源。

类型（？ c，容器）→类型（？ c，资源）

如果我们有 Type（A，容器），那么我们可以推断 Type（A，资源）

6.2.3.6 RDF 推断法

与 XML 不同，RDF 具有推理能力。虽然第一阶逻辑提供了一个证明系统，但是使用第一阶逻辑来开发 RDF 推理系统在计算上将是不可行的。因此，开发了霍恩子句（horn clause）逻辑（见本章参考文献［LLOY87］）用于逻辑编程，但是计算上仍然需要很高代价。语义 Web 基于一种限制逻辑，这种逻辑被称为描述性逻辑，其细节可以在本章参考文献［ANTO08］中找到。RDF 使用“如果那么”规则如下：

如果 E 包含三元组（？ u，rdfs：subClassof，？ v）和（？ v，rdfs：subClassof，？ w）

那么 E 还包含三元组（？ u，rdfs：subClassof，？ w）

也就是说，如果 u 是 v 的子类，v 是 w 的子类，则 u 是 w 的子类。

6.2.3.7　RDF 查询

像 XML 查询语言（如 X－Query 和 XML－QL）一样，适合于 RDF 的查询语言也正在开发中。虽然可以使用 XML 查询 RDF，但是这是非常困难的，因为 RDF 比 XML 更丰富。于是，RQL 被开发出来。RQL 是为 RDF 开发的类似 SQL 的语言。它的形式是：

Select from“RDF document” where some“condition.”

6.2.3.8　SPARQL

W3C 的 RDF 数据组为 RDF 开发了一种称为 SPARQL（Simple Protocol and RDF Query Language）的查询语言，现在它正在成为查询 RDF 文档的标准。我们正在为云计算开发 SPARQL 查询处理算法。另外，还开发了用于 SPARQL 查询的优化器。

6.2.4　本体

本体是任何实体、人或事物的通用定义。本体被用来阐明各种术语，因此对于机器可理解的网页至关重要。我们已经定义了几个可以使用的本体。这里的难题是如何定义一个实体的通用本体，因为不同的团体可能会出现不同的定义。因此，我们需要多个本体的映射。也就是说，这些映射将一个本体映射到另一个本体。我们已经为本体开发了专门的语言。请注意，类似 XML，所开发的 RDF，并不完全适合指定类/子类关系等语义。因为不能表达几种其他属性，如并集和交集，RDF 的功能也是有限的。因此，我们需要更丰富的语言。本体语言就是为这个目的由语义 Web 社区而开发。

OWL（Web 本体语言，Web Ontology Language）是一种流行的本体规范语言。它是本体的语言，依赖于 RDF。DARPA（国防高级研究计划署，Defense Advanced Research Projects Agency）开发了早期语言，DAML（DARPA 代理标记语言，DARPA Agent Markup Language）。欧洲人开发了 OIL（本体接口语言，Ontology Interface Language）。DAML＋OIL 结合起来，是 OWL 的雏形。OWL 由 W3C 开发。OWL 是基于 first－order 逻辑（第一阶逻辑）的子集，是一种描述逻辑。

OWL 特征包括：子类关系、类成员身份资格、类的等价性、分类和一致性（例如，x 是 A 的实例，A 是 B 的子类，x 不是 B 的实例）。

OWL 有 3 种类型：OWL－Full、OWL－DL、OWL－Lite。管理本体的自动化工具称为本体工程。

下面是 OWL 规范的一个例子

教科书和课本是一样的

英文书不是法文书

英文书不是德文书

```
<owl: Class rdf: about =“#EnglishBook”>
 <owl: disjointWith rdf: resource “#FrenchBook”/>
 <owl: disjointWith rdf: resource =#GermanBook”/>
</owl:Class>
<owl: Class rdf: ID =“TextBook”>
 <owl: equivalentClass rdf: resource =“CourseBook”/>
</owl: Class>
```

下面是 OWL 属性规范

英语书被学生阅读

```
<owl: ObjectProperty rdf: about = “#readBy”>
 <rdfs domain rdf: resource = “#EnglishBook”/>
 <rdfs: range rdf: resource = “#student”/>
<rdfs: subPropertyOf rdf: resource = #involves”/>
</owl: ObjectProperty>
```

下面是属性限制的 OWL 说明

所有的法语书仅被法语学生阅读

```
<owl: Class rdf: about = “#”FrenchBook”>
<rdfs: subClassOf>
 <owl: Restriction>
<owl: onProperty rdf: resource = “#readBy”>
<owl: allValuesFrom rdf: resource = #FrenchStudent”/>
</rdfs: subClassOf>
</owl: Class>
```

6.2.5 Web 规则 和 SWRL

6.2.5.1 Web 规则

RDF 建立在 XML 之上，OWL 建立在 RDF 上。我们可以在 RDF 中表达子类关系，其他关系可以用 OWL 表示。然而，OWL 的推理能力仍然有限。因此，我们需要指定规则和规则标记语言，以便机器可以理解并进行推理。

以下是本章参考文献［ANTO08］中给出的一些例子。

学生（X，Y），生活（X，Z），地点（Y，U），地点（Z，U）→国内学生（X）

也就是说，如果 John 在 UTD 学习，住在坎贝尔路，坎贝尔路和 UTD 的位置是理查德森，那么 John 是国内的学生。

需要注意的是，人（X）→男人（X）或女人（X）不是谓词逻辑中的规则。

如果 X 是一个人，那么 X 是男人或女人不能用一阶谓词逻辑来表达。因此，在谓词逻辑中，我们表达上述含义，就是：如果 X 是一个人，而 X 不是一个男人，那么 X 是一个女人。同样如果 X 是一个人，而 X 不是一个女人，那么 X 就是一个男人。也就是说，在谓词逻辑中，我们可以具有如下形式的规则

人（X）和非男人（X）→女人（X）。

但是，在 OWL 中，我们可以制定一个规则：如果 X 是一个人，则 X 是一个男人或 X 是一个女人。

规则可以是单调的或非单调的。

以下是一个单调规则的例子：

→母亲（X，Y）

母亲（X，Y）→父母（X，Y）

如果 Mary 是 John 的母亲，那么 Mary 是 John 的父母，

他的形式是：B1，B2，...，Bn→A

也就是说，如果 B1，B2，...，B*n* 成立，则 A 成立

在非单调推理的情况下，如果我们有 X 和 NOT X，我们将它们视为一致的单调推理。例如，如本章参考文献［ANTO08］中所讨论的，考虑 John 可以接受的公寓的例子。一般来说，John 准备租一套公寓，除非公寓不到两间卧室和不允许养宠物。这可以表示如下：

■ →可接受（X）

■ 卧室（X，Y），Y <2→不可接受（X）

■ 不允许养宠物（X）→不可接受（X）

第一条规定，公寓一般是 John 可接受的。第二条规定，如果公寓有不到两间卧室，那么 John 是不能接受的。第三条规定，如果不允许养宠物，那么 John 是不能接受的。请注意，这可能会有矛盾。但是，通过非单调推理，这是允许的，而这在单调推理中是不允许的。

我们需要规则标记语言让机器理解规则。逻辑的各种组件由被称为 RuleML 的规则标记语言进行表示，RuleML 是专门为语义 Web 开发的。单调和非单调规则都可以用 RuleML 表示。

例如，事实上的父母（A）的表述是“A 是父母”，其 RuleML 表示如下：

```
<fact>
        <atom>
    <predicate>Parent</predicate>
      <term>
        <const>A</const>
      </term>
    </atom>
</fact>
```

6.2.5.2　SWRL

W3C 提出了一种集成 OWL 和 Web 规则的新规则语言，这就是 SWRL。SWRL 的作者们指出，SWRL 对 OWL 公理集合进行了扩展，包括了 Horn－like 的规则。这样的话，Horn－like 的规则就可以与 OWL 知识库相结合。这样的语言将具有 OWL 的表达能力和逻辑编程的推理能力。我们在图 6.6 中说明了 SWRL 的组件。

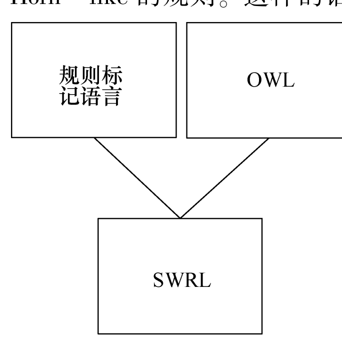


图 6.6　SWRL 组件

SWRL 的作者（Horrocks 等人）也指出，建议的规则以隐含的形式存在前项（body）和后项（head）之间。可以看出其中的含义是：无论任何时候，如果前项中指定了相应的限定条件，则后项中的限定条件也必须与之相同。针对这些规则，还给出了一套 XML 语法，当然这些语法也是基于 RuleML 和 OWL XML 表达语法。此外，我们还介绍了基于 OWL RDF / XML 交换语法的 RDF 具体语法。规则语法用几个运行示例说明。最后，我们给出使用建议和注意事项。

以下是我们从 SWRL（见本章参考文献［SWRL］）的 W3C 规范中获取的 SWRL 示

例。它指出，如果x1是x2的孩子，x3是x2的兄弟，那么x3是x1的叔叔。该示例使用了XML语法。有关SWRL的更多详细信息，请读者参考W3C规范［SWRL］。

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href = "#example1"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasParent">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="hasBrother">
      <ruleml:var>x2</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="hasUncle">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

6.2.6　语义 Web 服务

语义Web服务利用语义Web技术。正如我们在第5章中所述，WS使用基于XML的WSDL和SOAP消息。凭借语义Web技术，可以利用RDF在消息以及WS描述语言中表达语义。本体可以用来处理异质性。例如，如果消息或服务描述中的单词是模棱两可的，那么本体可以解决这些歧义。最后，规则语言（例如SWRL），可以用于消息的推理能力以及服务描述。

如本章参考文献［SWS］所述，用于WS互操作的主流XML标准，仅仅规定语法互操作性，而不是消息语义上的含义。例如，通过Web服务来发送和接收的数据结构，WSDL可以指定可用的操作，但它不能指定数据的语义含义或数据上的语义约束。如果需要说明语义含义，就要求程序员要达成关于WS交互的特定协议，并使自动Web服务组合变得困难。

语义Web服务是围绕语义Web标准构建的，用于交换语义数据，这使得编程人员可以轻松地将来自不同来源和服务的数据组合在一起，而不会失去意义。当Web浏览器向Web服务器发出请求时，WS可以被秘密激活，然后Web服务器使用各种WS来构建比本身能够做出更复杂的回复。语义Web服务也可以被自动化程序使用和运行，而无需连接到Web浏览器。

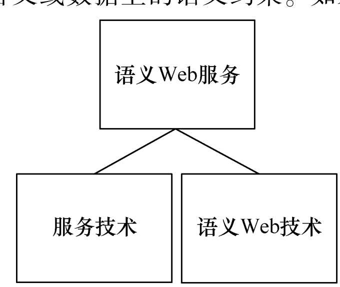


图6.7　语义Web服务

稍后在本书中，我们将讨论如何将语义Web、安全性和WS集成到语义Web服务中。图6.7说明了语义Web服务的各种组件。

6.3　安全的语义 Web 服务

6.3.1　语义 Web 的安全性

首先，我们概述语义 Web 的安全问题，然后讨论 XML 安全性、RDF 安全性和安全信息集成的一些细节，这些都是有关安全语义 Web 的组件。随着对各种问题研究的进一步发展，我们希望为确保语义网络安全开发适当的标准。安全性不能单独考虑，也就是说，没有一个层面应该集中关注安全性。安全性需跨越所有层次，这是一个挑战。总而言之，我们需要每个层都具有安全性，如图 6.8 所示。



图 6.8　安全语义 Web 层次

例如，考虑最低层。这一层需要安全的 TCP / IP、安全的套接字和安全的 HTTP。现在已经存在为各种低层协议开发的安全协议。用户需要端到端的安全性。也就是说，不能仅仅将安全的 TCP/IP 建立在不可信的通信层上，我们也需要网络安全。上一层是 XML 和 XML 架构。一个需要安全的 XML。也就是说，为了方便阅读、浏览和修改，访问必须被控制到文档的各个部分。有学者在从事 XML 和 XML 架构的安全研究。下一步是确保 RDF 安全。现在，RDF 不仅需要安全的 XML，还需要安全解释和语义。例如，在某些上下文中，文档的某些部分可能是非保密的，而在某些其他上下文中，文档可能是保密的。

一旦 XML 和 RDF 得到安全保护，下一步是检查本体和互操作的安全性。也就是说，本体可能具有属于它们自己的安全级别。本体的某些部分可能是保密的，而某些其他部分可能是非保密的。如何将这些本体用于安全的信息集成是一项挑战。研究人员对数据库的安全互操作性已经展开了一些研究。我们需要重新审视这项研究，然后确定还需要做些什么，以便可以安全地进行管理、集成和交换网络上的信息。逻辑、证据和可信是语义 Web 的最高层。那就是说，我们如何才能信任网络给我们的信息？

我们还需要检查语义 Web 的推理问题。推理是构成查询和推导新信息的过程。当推断的信息是用户未经授权获取的时，这会成为一个问题。使用语义 Web，特别是数据挖掘工具，可以做出各种推论。最近有一些关于在语义 Web 上控制未经授权推论的研究。我们需要继续进行这样的研究（参见本章参考文献［FARK03］和［THUR06］）。

安全不应该在事后才想到。我们经常听说，从一开始就需要把安全性植入到系统。类似地，安全性不能是对语义 Web 的事后考虑。有时我们在必须始终保证 100% 的安全性的同时，也不能使系统效率低下。更多的时候我们需要的是一个灵活的安全策略。在某些情况下，我们可能需要 100% 的安全性，而在某些其他情况下，30% 的安全性可能就足够了。

6.3.2 XML 安全性

有关 XML 安全性的各种研究工作已经有所报告（见本章参考文献［BERT02］)。我们这里简要讨论一些关键点。主要的挑战有是否允许访问整个 XML 文档还是部分文档。Bertino 等人已经开发了 XML 的授权模型，主要集中在访问控制策略以及发布策略。他们还考虑了推和拉的架构，也指定了 XML 中的策略。策略规范包含有关哪些用户可以访问哪些文档部分的信息。在本章参考文献［BERT02］中，介绍了访问控制算法以及结果的计算观点。此外，还讨论了用于保护 XML 文档的架构。在本章参考文献［BERT04］中，作者进一步介绍了如何在网络上发布 XML 文档。这么做，主要是为了拥有者发布文档，为了主体去请求访问这些文档，以及为不受信任的发布商在主体面前展现出被授权查看文档的假象。

W3C 正在制定 XML 安全性的标准。XML 安全项目（参见本章参考文献［XML1］)目前正着重实现 XML 安全标准。其重点是 XML 签名的语法和处理、XML 加密的语法和处理以及 XML 密钥管理。W3C 还有许多工作组，包括 XML 签名工作组（见本章参考文献［XML2］）和 XML 加密工作组（见本章参考文献［XML3］)。虽然我们正在关注其中近期可实施的标准内容，但是在保护 XML 文档方面还需要进行大量研究。

6.3.3 RDF 安全性

RDF 是语义 Web 的基础。虽然 XML 在提供机器可理解的文档方面受到限制，但 RDF 可解决此问题。因此，RDF 为互操作性以及搜索和编目提供了更好的支持。它还描述了文档的内容以及文档中各实体之间的关系。虽然 XML 仅提供语法和符号，但 RDF 通过以标准化的方式提供语义信息来补充这一点。

基本的 RDF 模型有 3 个组件：资源、属性和语句。资源是由 RDF 表达式描述的。它可以是一个网页或一组页面。属性是用于描述资源的特定性质。RDF 语句是资源与一个命名属性加上属性的值。语句组件是主体、谓词和客体。例如，如果我们有一个形式为“xxx 的创建者是 John”的句子，那么 xxx 是主体或资源，属性或谓词是“创建者”，客体就是文字“John”。类似 ER 图或对象图，RDF 图可以用来表示语句。重点是将预期的解释用于 RDF 语句，这是由 RDF 规范完成的。规范是一种字典，并且具有对语句中使用的各种术语解释的作用。

RDF 中更高级的概念包括容器模型和关于语句的声明。容器模型有 3 种类型的容器对象，它们是单元组、序列和待选内容。一个单元组是无序的资源或文字列表。它用来表示一个属性具有多个值，但顺序不重要。序列是有序资源的列表，这里的顺序很重要。待选内容是表示属性值的可选择的资源列表。关于 RDF 的各种教程更详细地描述了容器的语法。

RDF 还提供对其他语句声明的支持。利用这个功能，可以完成例如“语句 A 为假”的形式语句，其中 A 是语句“John 是 X 的创建者”。同样，可以使用对象类图来表示容器和关于语句的声明。RDF 也有一个与之相关联的正规模型，它有一个正规的语法。有关

RDF 的更多信息，可以参考 Antoniou 和 van Harmelen 的书中的精美论述［ANTO08］。

现在要使语义 Web 安全，需要确保 RDF 文档是安全的。这将涉及从语法角度来确保 XML 安全。然而，对于 RDF，我们还需要确保在语义层面的安全性，包括概念资源、属性和语句的安全隐患。如何保证访问控制？关于语句的声明和属性如何被保护？如何以粒度提供访问控制？容器模型的安全属性是什么？如何保护单元组、序列和待选内容？我们可以在 RDF 中指定安全策略吗？我们如何解决策略的语义不一致问题？我们如何在 RDF 中表达安全约束？陈述语句的安全隐患是什么？我们如何保护 RDF 架构？这些都是难题，我们需要开始研究来寻找答案。XML 安全只是一个开始，保护 RDF 更具挑战性（参见本章参考文献［CARM04］）。

6.3.4 安全性和本体

本体是为了避免歧义，对各种概念的本质性表示。我们已经开发了许多本体，这些本体已经被代理使用来理解网页并进行数据库的集成等操作。此外，本体可以用 RDF 或 OWL 等专用语言来表示。

现在，我们必须保证本体安全，就必须控制对本体的访问。这意味着不同的用户可能访问本体的不同部分。另一方面，本体可以用于指定安全策略，就像 XML 和 RDF 已被用于指定策略一样。我们将描述如何保护本体，以及如何使用本体来指定各种策略。

6.3.5 安全查询和规则处理

在 Secure RDF 层之上是安全查询和规则处理层。虽然 RDF 可以用于指定安全策略（参见本章参考文献［CARM04］），但 W3C 正在开发的 Web 规则语言对于指定复杂策略而言更为强大。此外，有学者正在开发推理机来对规则进行处理和推理（例如，在马里兰大学开发的 Pellet 推理机）。同样，也可以将我们开发的数据库推理控制器的想法（参见本章参考文献［THUR93］）与 Web 规则处理相结合，以开发语义 Web 的推理或隐私控制器。

查询处理模块负责访问语义 Web 上的异构数据和信息源。研究人员正在研究将 Web 查询处理技术与语义 Web 技术相结合的方法，以定位、查询和集成异构数据和信息源。另外，我们需要检查查询处理的安全性。

6.3.6 语义 Web 的隐私和信任

隐私是个人的相关需要保护的信息。另外，个人可向 Web 服务商指定具体关于他或她的可被公开的信息。特别是当涉及保护患者的医疗信息时，这样的隐私问题已经被大量讨论了。社会科学家和技术专家一直致力于隐私问题研究。然而，过去一年里，隐私受到了极大的关注。这主要是因为 Web 和语义 Web 的出现，以及反恐和国家安全问题。例如，为了提取关于各种人的信息，并且可能防止和/或侦测潜在的恐怖袭击，因此数据挖掘工具都要接受审查。我们在媒体上也听过很多关于国家安全与隐私的讨论。这主要是因为人们现在意识到要处理恐怖主义，政府可能需要收集有关个人的数据，并

挖掘数据来提取信息。数据可能在关系数据库中，也可能是文本、视频和图像。这些工作正在引起各种公民自由工会的重大关切（见本章参考文献［THUR02］）。与隐私密切相关的是匿名，有人认为保持匿名更重要。

通过 Web 和语义 Web，现在大量有关个人的数据可以在几秒钟内获得。数据可以是结构化数据，也可以是文本、图像、视频和音频等多媒体数据。信息可以通过采集或信息检索获得。数据挖掘是使网络更加智能的重要工具。也就是说，可以使用数据挖掘来挖掘网络上的数据，以便网络可以演化成语义网。但是，这也意味着隐私可能会受到威胁。因此，需要对语义网上的数据库和数据挖掘工具实施隐私控制。这是一个非常困难的问题。总之，需要开发技术来防止用户从网络或网络服务器上进行挖掘和提取信息。请注意，数据挖掘是对分析人员至关重要的技术，因为可以提取以前未知的模式。但是，我们不希望以不正确的方式使用信息。例如，根据相关个人信息，保险公司可以拒绝保险或贷款机构可以拒绝贷款。在许多情况下，这些否认可能不合理。因此，信息提供者必须谨慎对待他们要发布的信息。此外，数据挖掘研究人员必须确保隐私方面的问题得到解决。

虽然有关语义 Web 中隐私问题的研究工作报道很少，但是我们依然在朝着正确的方向发展。随着这一领域的主动性研究工作的开展，我们可以期待会有相应的进展。但是，我们还要考虑社会和政治方面。总之，技术专家、社会学家、策略专家、反恐专家和法律专家必须共同努力，开发适宜的数据挖掘技术以确保隐私。另外，还迫切需要隐私策略和标准。那就是说，当技术专家开发隐私解决方案时，我们需要决策者与标准组织合作，以便开发适当的隐私标准。W3C 在 P3P（隐私参数项目平台，Platform for Privacy Preferences）方面取得了良好的开端。

提供解决方案来加强国家安全，但同时要确保隐私，这是一个挑战。现在各种实验室都正在研究具有隐私增强/敏感/保护性质的数据挖掘技术（例如，IBM Almaden 的 Agrawal，康奈尔大学的 Gehrke 和普渡大学的 Clifton，详见本章参考文献［AGRA00］、［CLIF02］、［GEHR02］）。这里的想法是继续挖掘，但同时尽可能确保隐私。例如，Clifton 提出使用多方安全策略的方法进行隐私敏感数据挖掘。虽然有一些进展，但还有很长的路要走。部分有用的参考文献在本章参考文献［CLIF02］中提供（另见本章参考文献［EVFI02］）。

这里给出一些关于我们处理方法的细节。请注意，人们挖掘数据的目的是提取模式和趋势。隐私约束能确定哪些模式是隐私的，以及在多大程度上是隐私的。例如，假设可以提取名称和保健记录。如果我们有隐私约束规定姓名和医疗保健记录是隐私的，那么这些信息不向公众发布。如果信息是半隐私的，那么它将被发布给有需要知道的人。本质上，我们讨论的推理控制器方法是实现一定程度隐私的一个解决方案。它可以被认为是一种对隐私敏感的数据挖掘。在研究中，已经发现了我们开发的推理控制器方法中存在许多挑战（参见本章参考文献［THUR93］）。处理隐私约束时必须解决挑战（另见本章参考文献［THUR04］）。图 6.9 为语义 Web 的隐私控制器。如图 6.9 所示，在网络上有数据挖掘工具，用来挖掘 Web 数据库。隐私控制器应确保数据挖掘要基于隐私保

护。本体可由隐私控制器使用。例如，隐私结构可能存在本体规范。此外，可以扩展XML以包括隐私约束。RDF可以包含隐私语义。所以，我们还需要对隐私控制本体的作用进行更多的研究。

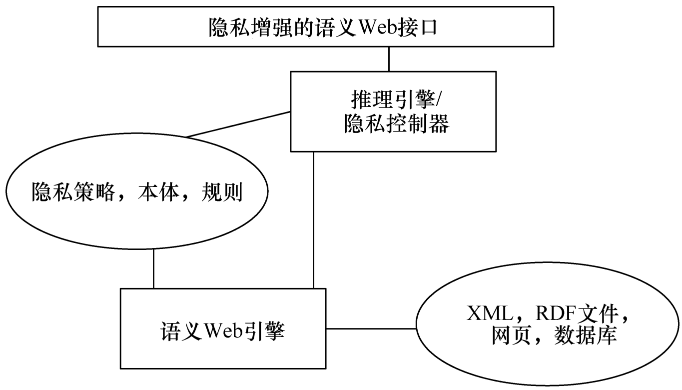


图6.9　语义Web的隐私控制器

关于隐私保护数据挖掘的大部分工作都侧重于关系型数据。我们需要对隐私保护语义Web的数据挖掘进行研究，也需将基于隐私保护的数据挖掘技术与语义Web数据挖掘技术相结合，以获得隐私保护语义Web的数据挖掘解决方案。

最近我们做了大量关于信任和语义Web上的研究工作（参见Finin等人的研究，见本章参考文献［DENK03］，［KAGA03］）。这里的难题包括如何信任网络上的信息？如何相信来源？如何在不同方之间进行协商并签订协约？如何将信任管理和协商的结构合并到XML和RDF中？信任管理的语义是什么？

研究人员正在制定信任管理协议。用于指定信任管理结构的语言及关于信任管理基础的研究也正在开展中。例如，如果A信任B和B信任C，那么A信任C？如何共享语义Web上的数据和信息，并保持自治性。信任管理存在很多要解决的问题，如何传递信任？例如，如果在50%的时间里A信任B，B在30%的时间里信任C，那么你为A信任C分配了多少信任时间？您如何将信任融入语义互操作性？信任和协商的服务原语的质量又是什么？也就是说，在某些情况下，可能需要100%的信任，而在某些其他情况下，50%的信任可能就足够了（另见本章参考文献［YU03］）。

而正在研究的另一个主题是信任传播和传播特权。例如，如果你向A授予权限，A可以转移到B什么权限？你如何组合特权？有为了特权组合而服务的代数和微积分吗？在此，仍需要进行更多研究。语义Web的一个层面是逻辑、证明和信任。在这个层面上，从根本上要解决不同代理商之间的信任管理和协商，并审查信任管理的基础和发展逻辑。

6.3.7　安全的语义Web和WS

WS与语义Web的集成造就了语义Web服务。也就是说，相对于WWW来说的WS就是相对于语义Web的语义Web服务。Tim Finin和他的团队讨论了语义Web服务架构（见本章参考文献［BURS05］）。他们描述了WS的不足之处，并论述了对语义Web服务的需求。说明了目前的技术允许使用WS，尤其当前的WS支持语法信息描述以及对

服务的发现、组合和执行的语法支持。他们认为人们需要语义标记的内容和服务，因此需要开发语义 Web 服务。然后，他们定义了一种称为语义 Web 服务的体系结构，该体系结构由一组体系结构和协议抽象构成，其作为语义 Web 服务技术的基础。这些技术支持如下功能：

- 动态服务发现；
- 服务参与；
- 服务流程制定，社区支持服务；
- 服务质量。

服务发现是客户识别候选服务的过程。协调者将服务请求者连接到提供商。可能需要本体来指定服务。服务合约指定了请求者和提供商之间的协议。因此，在此阶段进行约定协商。一旦服务准备好启动，服务发布阶段将开始。如本章参考文献［SHIB］所述，在此阶段，请求者确定请求服务所需的信息，和对服务成功或失败的相应反应。这也将包括解释响应和进行转换，社区管理服务支持认证和安全管理。服务质量为协商提供了支持，同时也为交付数据的安全性和及时性做出权衡。

安全性贯穿于所有这些服务。应注意到社区管理服务特别要求认证和安全管理，需要安全服务进行服务发现、引擎分段和发布。例如，并不是所有的服务都可以被发现。这取决于服务的敏感度和请求者拥有的安全凭证。因此，必须对语义 Web 服务描述，检查 XML、RDF 和 OWL 的安全性规范。图 6.10 为安全的语义 Web 服务的构成。

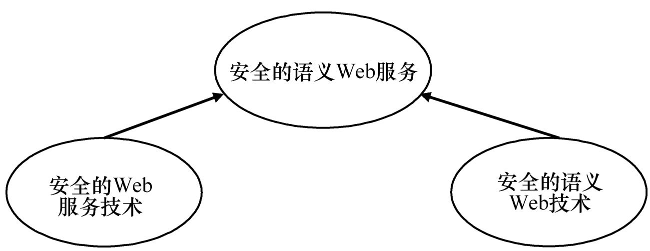


图 6.10　安全的语义 Web 服务

6.4　总结和展望

本章首先提供了语义 Web 技术的概述和语义 Web 服务的概念。特别是，我们讨论了 Tim Berners Lee 的技术方案。然后我们讨论了 XML、RDF、本体以及语义 Web 的 WS。最后，我们讨论了语义 WS 以及如何利用语义 Web 技术。请注意，就像 WS 是云的核心一样，语义 Web 服务是语义云的核心。如前所述，语义云是利用语义 Web 技术的云。也就是说，它们是提供语义服务的集合。

接着，我们概述了语义 Web，并讨论了安全标准。讨论了语义 Web 的安全问题，我们认为安全必须贯穿所有层次。接下来，我们提供了有关 XML 安全性、RDF 安全性、安全信息集成和信任的更多细节。如果要保证语义 Web 是安全的，我们就需要保证所有的组件都是安全的。我们还介绍了对访问控制和 XML 文档传播的一些研究。最后，

讨论了语义 Web 的隐私问题。

在语义 Web 服务上还有很多工作要做。WS 的大部分开发都集中在 XML 技术上。我们需要制定 WS 版本的 RDF 标准。例如，我们需要为 WS 描述开发类似 RDF 的语言。有关语义 Web 服务研究才刚刚开始。安全语义 Web 服务本质上集成了语义 Web 服务技术和安全技术。

在确保语义 Web 安全方面还需要做许多研究。需要继续研究 XML 安全性，还必须审查 RDF 的安全性。由于 RDF 融合了语义，所以更为困难。我们需要审查安全约束处理和上下文相关的安全约束工作，看看我们是否可以为 RDF 的安全性采取一些措施。之后，也需要研究本体对安全信息集成的作用。我们必须解决一些困难的问题，比如，如何将安全策略整合到语义 Web 上？如何将策略纳入本体？另外，我们也不能忘记语义 Web 的隐私和信任。也就是说，我们需要保护个人的隐私，同时确保个人为了执行职能，而拥有所需的信息。最后，需要形式化信任的概念，并研究在语义 Web 上协商信任的方法。我们有一个很好的开始，并且很好地建立了语义 Web。需要注意的是，我们必须在开始时就考虑安全性，而不是事后考虑。

标准在语义 Web 的发展中起着重要的作用。W3C 已经非常高效地制定了 XML 和 RDF 的标准。我们需要继续发展，尽可能地将研究成果转化到标准中。我们还需要将研究成果和标准转移到商业产品中。语义 Web 标准的下一步工作，是检查安全性、隐私性、服务质量、完整性以及其他功能（如安全查询服务）。正如我们强调的那样，安全和隐私是至关重要的，必须在制定标准的同时进行调查研究。

我们对语义 Web 和安全语义 Web 服务进行了概述，因为如果要提供机器可理解的服务，云服务需要使用语义 Web 服务。此外，我们为云和云安全设计了几个实验系统，它们已经使用了语义 Web 技术。我们称这种云服务是语义云服务。第四部分、第六部分和第七部分将讨论这些实验系统。

参 考 文 献

[AGRA00] Agrawal, R. and R. Srikant. Privacy-preserving data mining, *Proceedings of the ACM SIGMOD Conference*, Dallas, TX, May 2000.

[ANTO08] Antoniou, G. and F. van Harmelen. *A Semantic Web Primer*, MIT Press, Cambridge, MA, 2008 (2nd Edition).

[BERT02] Bertino, E. et al. Access control for XML documents. *Data and Knowledge Engineering*, 43(): 19–34, 2002.

[BERT04] Bertino, E., B. Carminati, E. Ferrari, B. Thuraisingham, and A. Gupta. Selective and authentic third party distribution of XML documents. *IEEE Transactions on Knowledge and Data Engineering*, 161(10): 1263–1278, October 2004.

[BURS05] Burstein, M. H., C. Bussler, M. Zaremba, T. W. Finin, M. N. Huhns, M. Paolucci, A. P. Sheth, and S. K. Williams. A semantic web services architecture. *IEEE Internet Computing*, 9(5): 72–81, 2005.

[CARM04] Carminati, B., E. Ferrari, and B. Thuraisingham, Using RDF for policy specification and enforcement, *Proceedings of the DEXA Conference Workshop on Web Semantics*, Zaragoza, Spain, 2004, pp. 163–167.

[CLIF02] Clifton, C., M. Kantarcioglu, and J. Vaidya. *Data Mining: Next Generation Challenges and Future Directions*, H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds), Defining Privacy for Data Mining, AAAI/MIT Press 2004, pp. 255–272.
[DENK03] Denker, G., L. Kagal, T. Finin, and M. Paolucci, Security for DAML web services: Annotation and matchmaking, *International Semantic Web Conference*, Sanibel Island, FL, USA, 2003.
[EVFI02] Evfimievski, A., R. Srikant, R. Agrawal, and J. Gehrke, Privacy preserving mining of association rules, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada, July 2002.
[FARK03] Farkas, C. et al., Inference Problem for the Semantic Web, *Proceedings of the IFIP Conference on Data and Applications Security*, Colorado, August 2003 (formal proceedings published by Kluwer, 2004).
[GEHR02] Gehrke, J., Research problems in data stream processing and privacy-preserving data mining, *Proceedings of the Next Generation Data Mining Workshop*, Baltimore, MD, November 2002.
[HEND01] Hendler, J., Agents and the semantic web, *IEEE Intelligent Systems Journal*, Match 2001.
[KAGA03] Kagal, L., T. W. Finin, and A. Joshi, A policy based approach to security for the semantic web. *International Semantic Web Conference* 2003, Sanibel Island, FL, USA, pp. 402–418.
[LAUR00] St. Laurent, S., *XML: A Primer*, Power Books Publishing, UK, 2000.
[LEE01] Lee, T. B. and J. Hendler, The semantic web, *Scientific American*, May 2001.
[LLOY87] Lloyd, J. *Logic Programming*, Springer, Germany. 1987.
[SHIB] Shibboleth Internet 2, http://en.wikipedia.org/wiki/Shibboleth_%28Internet2%29
[SWS] Semantic web services, http://en.wikipedia.org/wiki/Semantic_Web_Services
[SWRL] World Wide Web Consortium, http://www.w3.org/Submission/SWRL/#1
[THUR93] Thuraisingham, B. et al. Design and implementation of a database inference controller. *Data and Knowledge Engineering Journal*, 11(3), 271–297, 1993.
[THUR02] Thuraisingham, B. M. Data mining, national security, privacy and civil liberties. *SIGKDD Explorations*, 4(2), 1–5, 2002.
[THUR04] Thuraisingham, B. Privacy constraint processing in a privacy enhanced database system. *Data and Knowledge Engineering Journal*, 55(2), 159–188, 2005.
[THUR05] Thuraisingham, B. *Database and Applications Security: Integrating Data Management and Information Security*, CRC Press, Boca Raton, FL, 2006.
[THUR06] Thuraisingham, B. et al. Administering the semantic web, confidentiality, privacy and trust. *Journal of Information Security and Privacy*, 2006.
[W3C] The World Wide Web Consortium, www.w3c.org
[XML1] The Apache Foundation, http://xml.apache.org/security/
[XML2] The World Wide Web Consortium, http://www.w3.org/Signature/
[XML3] The World Wide Web Consortium, http://www.w3.org/Encryption/2001/
[YU03] Yu, T. and M. Winslett, A unified scheme for resource protection in automated trust negotiation, *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003, pp. 110–122.

第 7 章　专用的 Web 服务和安全

7.1　概述

第 4 章论述了数据、信息和知识管理中的一些基本概念，第 5 章论述了 WS 的安全性，第 6 章论述了语义 WS 和安全性，在本章中，我们将论述专用的 WS 及相关的安全问题。专用 WS 包括数据和信息以及知识管理的服务，信息互操作性和电子商务等活动管理 WS，医疗保健和金融等领域行业的 WS。

我们首先论述数据管理、信息管理、知识管理和活动管理等专用 WS 的一些细节。我们还将论述域 WS 和软件以及数据作为服务的概念。然后，我们分别论述这些主题的安全性。例如，正如讨论安全域 WS、软件和数据作为服务的安全问题一样，我们将论述安全性对于数据、信息、知识和活动管理的影响。我们还将第 6 章论述的语义 Web 技术整合到这些 WS 中。

本章的结构如下：在 7.2 节中，我们将论述数据、信息和知识以及活动管理等专用的 WS。这些 WS 的安全性将在 7.3 节中论述。本章结束于第 7.4 节。图 7.1 为本章论述的概念。正如我们之前所论述的，WS 是云计算的核心。因此，数据、信息、知识和活动管理的 WS 是云数据、信息、知识和活动管理的实质。例如，云数据管理服务（例如查询服务）将利用本章论述的数据管理服务。第三部分和第五部分将论述云数据、信息、知识和活动管理的 WS。第四部分、第五部分和第七部分论述了与这种基于云的 WS 有关的实验系统。

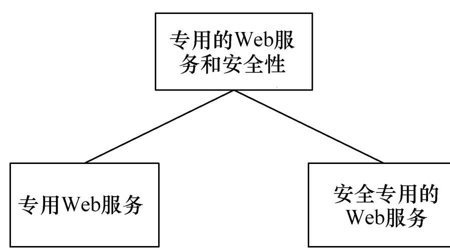


图 7.1　专业化 Web 服务和安全性

7.2　专用 Web 服务

7.2.1　总述

在 7.2 节中，我们将论述各种专用的 WS。这些包括数据、信息和知识管理的 WS 以及域 WS。这些 WS 使用第 5 章中论述的标准，如 SOAP、XML、UDDI 和 WSDL。

本节的结构如下：数据管理的 WS 将在 7.2.2 节中讨论；关于地理空间数据管理等复杂数据管理的 WS 将在 7.2.3 节中讨论；信息管理的 WS 将在 7.2.4 节讨论；第 7.2.5

节将讨论知识管理的 WS；第 7.2.6 节将讨论电子商务和信息共享等活动管理的 WS；域 WS 将在 7.2.7 节中讨论；一些新兴的 WS 将在 7.2.8 节中讨论。

7.2.2 数据管理的 Web 服务

各种数据管理功能可以作为 WS 被调用。例如，查询 WS 将包括多个 WS 的组合，查询修改服务和查询优化服务。查询服务将调用存储服务以从存储中检索数据。事务服务将执行事务。

WS 也可以使用语义 Web 技术来产生语义 WS。起初，一些策略（例如，管理策略和完整性策略）可以用 XML 和 RDF 之类的语言表示。这是语义 Web 的重要贡献之一。现在，数据库也可能由 XML 和 RDF 文件组成。例如，Oracle 公司的产品现在具有管理 XML 和 RDF 文档的能力。因此，我们需要应用数据管理技术来管理 XML 和 RDF 文档。

语义 Web 技术在异构数据库集成中具有应用。例如，需要本体来处理语义异质性。XML 现在被用作公共数据表示语言。关于数据仓库，可以使用 XML 和 RDF 来指定策略。此外，本体还可用于数据转换，以将数据导入到仓库。因为本体能阐明各种概念促进数据挖掘，使得它也能应用于数据挖掘。另一方面，网络上的大量数据将不得不被挖掘出来，以提取信息来引导代理理解网页。总之，在数据管理的各个方面，WS 和语义 Web 技术都有应用。图 7.2 说明了数据管理的 WS。



图 7.2　数据管理的 Web 服务

7.2.3 复杂的数据管理 Web 服务

多媒体和地理空间数据管理操作（如查询）可以作为 WS 调用。此外，XML 正在扩展，以便用于多媒体和地理空间数据。例如，SMIL（同步多媒体集成语言，Synchronized Multimedia Integration Language）是视频标记语言，而语音是音频数据的标记语言。在以 SMIL 表示的视频数据上，可以执行以 XML 和 RDF 或者更多其他描述性语言（如 REI）指定的访问控制策略。OGC（开放地理空间信息联盟，Open Geospatial Consortium）等组织制定了 GML（地理标记语言，Geography Markup Language）。OGC 规定了作为数据层编码的地理空间标准，这个标准就依赖于 GML 制定。OWL－S 提供了语义丰富的应用级平台，用于使用描述性逻辑对 Web 服务元数据进行编码。作为地理空间语义 Web 互操作性实验的一部分，OGC 正在寻找新的路径来整合 SMIL 和 GML。图 7.3 说明了复杂数据管理的 WS（例如地理空间数据管理）。在这里，服务提供商在 GML 或 GRDF（我们设计的地理空间 RDF）中发布服务，客户端将从目录中获取服务的位置，然后从服务提供商获取服务。

我们对 RDF 进行了扩展，形成了 GRDF，以适用于地理空间数据。我们使用 GRDF

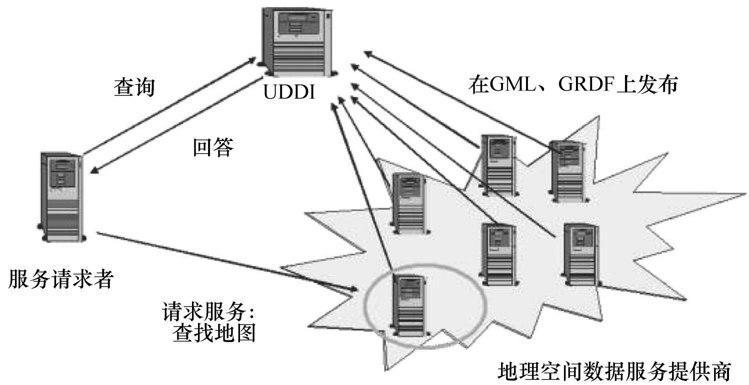


图 7.3　复杂数据管理的 Web 服务

中指定的本体来处理语义异质性。这种本体然后被用于语义互操作性。GRDF 的详细信息可以在本章参考文献［ALAM06］中找到。在 GRDF 之上，我们开发了地理空间本体。本体的描述在本章参考文献［THUR07］中。我们的团队（Thuraisingham、Ashraful、Subbiah 和 Khan）已经开发了一个名为 DAGIS 的系统，主要用于本体和答案查询。该系统在本章参考文献［THUR07］中有描述，它是一个框架，提供了一种在地理空间数据编码级别和服务框架上实现语义互操作性的方法。DAGIS 是一个集成平台，提供使用 OWL－S 服务本体，构建地理空间数据交换接口的机制和架构。为了本体自动发现，动态组合和服务调用结合地理空间领域特征，DAGIS 是获取和整合地理空间数据的一站式平台。借助 GRDF 中的数据编码，以及通过 DAGIS 或客户端代理提供智能推理，可以推理出有效载荷数据的能力。服务级别的 DAGIS 和数据编码层的 GRDF 提供了一个完整的统一模型，以便实现地理空间语义 Web 的愿景。该架构还增强了对 DAGIS 接口提出的客户机查询的响应功能。图 7.4 为 DAGIS 架构。

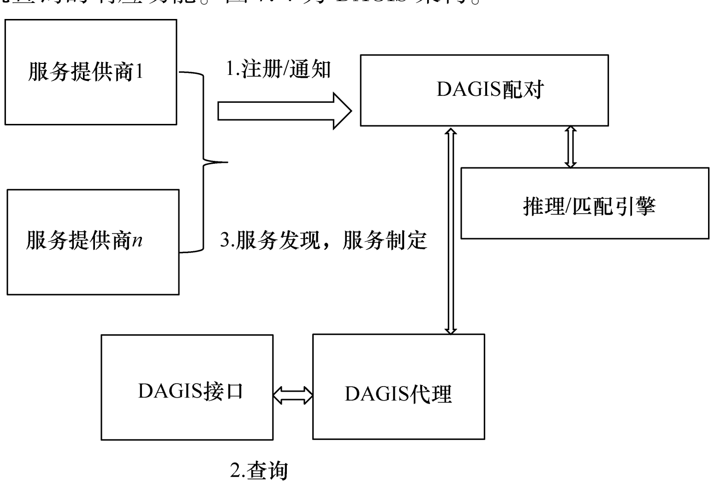


图 7.4　DAGIS 架构

围绕关于地理空间数据的 WS，其他成果还包括 GeoRSS，它是一种地理空间语义 Web 技术的应用，我们将在本节中讨论。如本章参考文献［OGC］所述，GeoRSS 是一个简单的建议，用于地理启用或标注，“简易信息聚合”（Really Simple Syndication，RSS）能提供位置信息。GeoRSS 提出了一种标准化的方式，其中定位编码具有足够的简单性和描述性，以满足大多数网页内容描述位置的需要。GeoRSS 也是一种轻量级的方式，以其他基于 XML 的格式（包括 XHTML）来描述地理信息。

7.2.4　信息管理的 Web 服务

WS 可以被各种信息管理应用程序调用。例如，数据挖掘和数据仓库操作可以作为 WS 实现。类似地，信息检索操作以及数字图书馆管理操作也可以作为 WS 被调用。如数据管理一样，诸如 XML、RDF 和 OWL 之类的语义 Web 技术，可用于表示各种管理和完整性策略。

语义 Web 技术也被应用于信息管理，特别是数字图书馆管理。Studer 及其团队（见本章参考文献［STUD］）提出了数字图书馆应用语义 Web 的重要方向。他们指出，“数字图书馆中语义技术的典型使用场景包括用户接口和人机交互（显示信息，允许大型信息收集的可视化和导航）、用户概况（考虑到整体信息空间）、个性化（平衡于基于个人和社区之间的个性化）、用户交互”。他们阐述了 SEKT 项目，试图解决许多难题。他们进一步指出，通过使用本体和语义 Web 技术，将会建成一些数字图书馆，他们能够给出对数字图书馆的一致看法。我们总结了他们在论文中讨论的一些对数字图书馆有用的先进语义 Web 技术。它们是本体论、本体编辑器、注释工具和推理引擎。有关详细信息，请参见本章参考文献［THUE10］。图 7.5 为信息管理的 WS。



图 7.5　信息管理的 WS

7.2.5　知识管理的 Web 服务

各种知识管理操作，比如创建和管理知识产权、存储和管理公司的专业知识以及维护公司网站等，都可以作为 WS 被调用。这些 WS 可以利用语义 Web 技术。

语义 Web 技术在知识管理中有许多应用。例如，我们需要本体来捕获和表示知识和理论。在关于语义 Web 和知识管理的文章中，保罗·沃伦（Paul Warren）举了一个例子，是关于“一个政治学家萨莉，她想研究英国首相托尼·布莱尔对津巴布韦的态度在一年多的时间里发生了多大程度的变化，以及是什么因素导致了这种变化”，她该如何研究这些？他进一步指出，“在语义 Web 的世界里，萨莉可以在一个特定的时期内，搜索布莱尔写的关于这个话题的一切文稿。她也可以搜索他的演讲手稿。信息标记

不会因文章或报告级别而停止，而标记也将存在于文章部分中。所以，萨利还可以找到政治评论家写的其中包含布莱尔演讲手稿的文章”（见本章参考文献［WARR06］）。

现在，知识管理也有一些应用需要建立语义 Web。例如，作为知识管理的结果，捕获的现有知识可以由代理使用，以更好地理解网页。图 7.6 为知识管理的 WS。

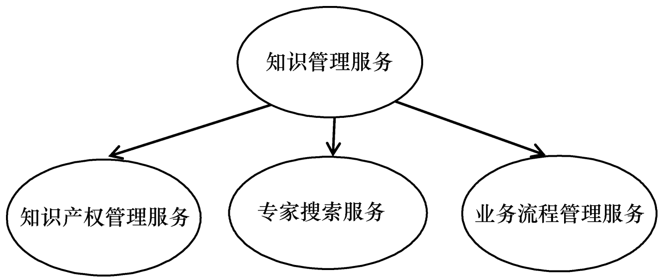


图 7.6　知识管理的 WS

7.2.6　活动管理的 Web 服务

在第 4 章中，我们讨论了活动管理，包括电子商务、信息共享和信息互操作性。在本节中，我们将讨论 WS 的活动管理。图 7.7 说明了 WS 与活动管理之间的关系。我们将讨论每个活动的细节。

7.2.6.1　电子商务

WS 和语义 Web 已经应用于电子商务的多个方面。一是为电子商务应用开发专门的标记语言，例如电子商务可扩展标记语言（Electronic Business using eXtensible Markup Language，ebXML）；二是语义电子商务，也就是利用语义 Web 技术实现的电子商务流程；三是使用 WS 来调用电子商务应用。



图 7.7　用于活动管理的 Web 服务

如本章参考文献［EBXM］所述，ebXML 是由 OASIS 和 UN / CEFACT（联合国贸易便利化和电子商务中心）发起的基于 XML 的标准家族，其任务是提供一个开放的基于 XML 的基础设施，其使全球所有贸易伙伴可以互操作，采用安全和一致的方式使用全球电子商务信息。该项目的初始目标是为业务流程制定 XML 标准。这些标准包括合作协议、核心数据组件、消息传递、注册表和存储库。

针对电子商务应用，开发了相关本体，这些本体以 RDF、RDF－S、OWL 和 OWL－S 等语言定义。例如，在 OBELIX 项目中，提出了对电子商务和本体非常好的描述。作者指出，电子商务的问题是这些模糊概念导致缺乏精确的描述。然后，他们根据需求工程来阐述他们的方法，并且定义了电子商务的本体。

该项目的更多细节在本章参考文献［OBEL］中给出。它说明，“OBELIX 是世界上

第一个基于本体的电子商务系统，提供智能、可扩展的集成和互操作性功能。”他们还表示，该项目“结合本体管理和配置、电子商务、应用服务器和基于本体的电子应用工具以及电子商务图书馆。“OBELIX 是一个欧洲委员会的项目，其目标是在语义 Web 环境中自动地提供电子商务服务，这种语义 Web 环境已经被称为语义电子商务。

一些关于语义电子商务的有趣的工作，正在由北卡罗来纳州格林斯博罗大学的一个小组进行研究。他们表示，语义电子商务是一些组织机构利用公司知识，协调地设计业务流程（见本章参考文献［SING06］）。它本质上将语义 Web 技术与业务流程管理和知识管理相结合。业务流程利用知识管理来提高其效率和效用，并使用语义 Web 技术（如本体）方便更好地理解。

语义商务，也就是一种或多或少的语义电子商务，也正在研究中。例如，位于布里斯托尔的惠普实验室的研究人员提出了企业对企业的电子商务交互的生命周期，并展示了语义 Web 如何支持服务描述语言，这种服务描述语言可在整个生命周期中使用。他们表明，通过使用 DAML + OIL，能够开发一种服务描述语言，不仅可以用于描述广告，还可以实现配对查询、协商建议和协议（见本章参考文献［TRAS］）。

7.2.6.2 协作与工作流

语义 Web 技术也可以应用于工作流和协作应用。例如，工作流管理联盟开发了两种语言。第一种是 Wf – XML（工作流 XML，Workflow XML），如本章参考文献［WFMC］所述，“Wf – XML 扩展了 ASAP（OASIS 的异步服务访问协议）模型，包括 BPM（Business Process Management，业务流程管理）和工作流交换功能。这个联盟还开发了 XPDL（XML Process Definition Language，XML 过程定义语言）。如本章参考文献［WFMC］所述，“XPDL 为实现业务流程管理和工作流引擎，以及设计、分析和交换业务流程提供了一个框架。

从文字上来说，虽然我们在这里讨论标记语言与 XML 相当，但是这些语言已经用本体扩展，为多媒体、工作流和协作计算应用程序提供语义。例如，苏格兰研究人员已经开发了基于 RDF 的语言，应用到了协作和工作流应用程序（见本章参考文献［CHEN04］）。

7.2.6.3 信息集成

虽然语义 Web 技术是为机器可理解的网页开发的，特别是 XML 被开发用于在 Web 上进行文档交换，但这些技术也可以扩展应用到信息互操作性方面。20 世纪 90 年代，语法异质性是一个主要问题，比如数据模型异质性。各网络技术社区正在讨论通用对象模型的开发，和面向通用数据表示的扩展关系模型（见本章参考文献［THUR97］）。然而，由于 XML 的发展，它是全球数据表示的首选语言。许多组织，包括国防部（Department of Defense，DoD）正在使用 XML 和 XML 框架来发布各个数据库的元数据。这是对通用数据模型的重大发展。

虽然 XML 是表示语法的理想选择，但我们需要用于表示语义的 RDF 和类似 OWL 的语言。因此，基于 RDF 的语言正在被用来处理语义异质性。例如，本体被用来定义各种术语，以及表示通用语义或区分不同的语义。然后，这些本体就用于信息互操作

性，并理解不同组织之间的各种术语。

7.2.6.4　其他活动

组织可以调用 WS 用于信息共享、社交网络和供应链管理。例如，组织 A 可以调用一个 Web 服务将相关信息放置到共享空间中，另一个 Web 服务可以检索到另一个组织放置的信息。组织可以通过调用 Web 服务来向提供商请求部分内容。该 Web 服务可以调用其他 WS 来请求其他部分内容。

语义 Web 技术也可用于信息共享、社交网络和供应链管理。例如，要共享的信息可以用 XML、RDF 或 OWL 表示。基于语义 Web 技术的框架可以用作信息共享的平台。在社交网络的情况下，可以从数据中提取第一本体，以形成社交网络。这些本体可以被用来提取模式。语义 Web 社交网络的一个很好的例子是在 RDF 中指定的 FOAF（朋友之友，Friend Of A Friend，FOAF）。

7.2.7　领域 Web 服务

在本节中，我们将讨论产业领域的 WS，包括防御、国土安全和医疗卫生。图 7.8 为我们将要探讨的各个行业。

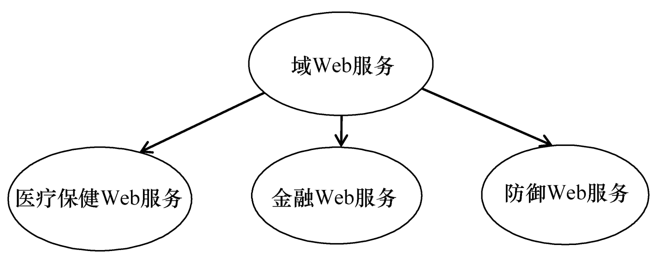


图 7.8　领域 Web 服务

7.2.7.1　防御

最早使用 WS 的领域就是防御和信息领域。在 Jim Handler 博士的管理下，DARPA 的 DAML 计划为国防部开发了相关技术。虽然安全性在该程序中不是考虑因素，但是它开发了一种称为 DAML（DARPA Agent Markup Language，美国国防高级设计研究署代理置标语言）的本体语言。该计划与 W3C 紧密合作，开发了机器可理解网页的技术。然后，DAML 与欧洲标准 OIL（本体接口语言，Ontology Interface Language）集成，以开发 DAML + OIL。当美国和欧洲共同开发 DAML + OIL 时，W3C 开发了 OWL 本体论。如前所述，OWL 从 RDF、DAML 和 OIL 发展而来。除了表示数据之外，关于数据的推理也是 DAML 计划的重点。

在 20 世纪 90 年代末和 21 世纪初，关于 DAML 计划进行的这段时间里，美国国防部参与了 GCCS（全球指挥和控制系统，Global Command and Control System）计划的发展。在这个方案下，开发了 DII COE（国防信息基础设施通用作战环境，Defense Information Infrastructure Common Operating Environment）。DII COE 基本上由几个工作组组成，包括用于分布式计算系统、多媒体和数据管理的工作组。然而，随着 WS 的出现，美国国防部开始大量投资以网络为中心的企业服务（Network – Centric Enterprise Services，

NCES），以达到网络为中心的运营。这导致了全球信息网格的发展，这基本上也是 NCES 的基础设施。这个基础设施是基于面向服务的架构和 WS。因此，它的许多发展受到 XML 和本体的影响。此外，兴趣社区（Communities Of Interest，COI）已经形成，这些社区已经为其应用开发了共同的本体。WS 也在国土安全应用中发挥重要作用。例如，本章参考文献［PALI04］中已经探讨了使用语义 Web 技术和 WS 来实施 RFID（射频识别，Radio Frequency Identifcation）标签。

7.2.7.2 医疗保健与生命科学

W3C 在 2004 年 10 月举办了一场研讨会，汇集了生命科学研究人员，以确定如何利用语义 Web 技术。今天，一些专家正在专注于为医疗保健和生命科学应用开发本体、WS 和标记语言。例如，本体用于指定药物和各种医学术语。例如，作为 W3C 中 Web 本体工作组的成员，Jonathan Borden 已经为医疗保健应用程序指定了 XML（见本章参考文献［BORD］）。他表示，他的目标是使用本体和标记语言来回答，比如“1997－2000 年期间所有我给做过脑肿瘤的患者中，匹配病理严重程度和匹配临床状态，并且具有 P53 突变的患者，PCV 化疗在五年内是否提高了治愈率?”他随后说明如何利用 XML、RDF 和 OWL 来有效地回答这些问题。

本体已经在电子医疗记录以及生命科学中的几个项目中被开发出来（见本章参考文献［KREM］）。例如，作者在本章参考文献［LIFE］中指出，“当代生命科学研究包括的各种组成部分，都是从物理、化学、数学、医学和许多其他领域中吸收而来的，所有这些方面以及基本的哲学问题，在构建域本体中必须要考虑到。”然后，他介绍了如何开发生命科学的域本体。

7.2.7.3 金融

金融领域包括必须处理金融的任何领域，包括银行和交易、保险和投资管理。几乎所有这些活动现在都以电子方式进行。我们现有电子交易、电子银行和电子保险管理等。在本节中，我们将研究可信赖的语义 Web 在金融领域的应用。

一些小组正在为金融领域开发 WS 和语义 Web 技术。例如，马德里的小组对应用语义“本体论”平台做了一些非常好的研究，该平台提供了①对内容和语义的集成，该集成以知识基方式体现，基于低级内容提供概念性观点，②自适应的基于超媒体的知识可视化和导航系统及③语义搜索设施（Semantic Search Facilities，CAST）（见本章参考文献［CAST］）。此外，他们已经开发出经济和金融信息的拓扑。比利时的另一个小组正在开发金融安全欺诈检测的本体。他们在项目中使用基于本体的知识工程来检测金融安全欺诈。特别是，他们从法规和法律上制定了欺诈司法本体（见本章参考文献［ZHAO］）。

另外，针对特定的项目，如在马德里和比利时的工作组中，为金融数据管理开发本体和语义，XML 被广泛用于金融服务，现在这被认为是金融的标准。如本章参考文献［XML］所述，“金融服务行业正在创建各种标准 XML 格式以满足其特殊需求”。正在开发的标准清单包括以下内容：

互动金融交换（Interactive Financial eXchange，IFX）和开放式金融交换（Open Fi-

nancial eXchange，OFX)，用来处理消费者和其他形式的零售银行业务。

金融信息交换（Financial Information eXchange，FIX）正在成为股票交易数据的标准通信协议。

FIX 标记语言（FIX Markup Language，FIXML）使用 XML 来表达 FIX 协议的业务消息。

金融产品标记语言（Financial products Markup Language，FpML）是一种基于 XML 的交换格式，用于金融衍生品市场的交易。

市场数据定义语言（Market Data Defnition Language，MDDL）是市场数据的定义和通信的联盟标准，它基于 XML，这些数据包括用于分析、交易和评估市场价值的数据，他们要借助金融工具来处理。

可扩展商业报告语言（eXtensible Business Reporting Language，XBRL）是一个“基于 XML 的规范，用于准备和交换财务报告以及数据”，它由组织和机构的全球联盟开发。

7.2.7.4　电信

WS 的另一个领域应用是电信。像爱立信、诺基亚和 AT&T 这样的公司正在为这个行业开发 WS。Parlay X 是 Parlay 集团的一项成就。如维基百科的文章所述，“Parlay 集团是一个技术行业联盟（成立于 1998 年)，为通信网络指定了 API（应用程序编程接口)。这些 API 使得传统运营商内部和外部的组织都能够创建服务。“2003 年，该集团开发了一套新的名为 Parlay X 的 WS，这是一组开发人员使用的更简单的 API。”如本章参考文献［TELE］所述，“Parlay X Web 服务包括第三方呼叫控制（Tird – Party Call Control，3PCC)、定位和简单支付等功能。”

本章参考文献［TELE］中提供了一个关于备受关注和有用的电信业 WS 的调查。文章指出，由于标准的变化、竞争和技术进步，近年来电信业一直处于崛起状态。作者为这个行业使用 WS 提供了强有力的例证，并解释了 Parlay X 的事件驱动架构和开发技术。

7.2.8　新兴的 Web 服务

7.2.8.1　X 即服务

越来越受欢迎的是使用 X 作为服务。X 可以是数据、软件平台、基础设施或任何受人关注的东西。使用数据作为服务，组织可以利用数据提供者，获取数据并调用数据作为服务。在软件作为服务的情况下，组织可以从服务提供商获得作为服务的编译器、操作系统或应用程序。在本节中，我们将说明这些服务。一般来说，X 作为服务由 XaaS（X as a Service）表示。

7.2.8.2　数据即服务

数据即服务（Data as a Service，DaaS）已经存在了相当一段时间。例如，Choice Point 和 Acxiom 等公司管理金融和医疗行业的各种公司的数据。数据服务可能包括数据安全和隐私服务，以及数据质量和清理服务。图 7.9 为作为服务模型的数据。尽管

DaaS 也被用于桌面服务，但我们这里将数据作为服务称为 DaaS。

将数据服务与 Web 服务技术集成是最近的一个概念。如本章参考文献［DATA］中所述，一旦您通过新的、能够返回您发送或执行的字符串 WS 时（比如说数学计算），那么服务就要促进完成相应的数据的插入或检索。无论您是要通过服务，请求检索客户和产品数据，还是想要将关键业务合作伙伴的供应链操作公开，关键都是要将数据访问层折合成 SOA。WSO2 数据服务为您的数据服务定位提供了一个方便和精心设计的机制。

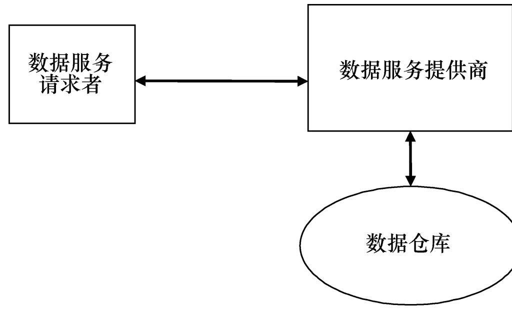


图 7.9　数据即服务

WSO2 企业服务总线允许服务松散耦合，以受管理的虚拟化方式连接系统，使管理员能够控制和直接通信，而不会中断现有的应用程序。WSO2 具有许多组件，数据服务器组件本质上是提供数据服务，包括集成各种数据源、数据库管理和相关服务等。

正在发展的另一个概念是使用数据库管理作为一项服务。加利福尼亚大学尔湾分校的 Sharad Mehrotra 和他的团队与 IBM、普渡大学和得克萨斯大学达拉斯分校的研究人员一起对这一概念进行了研究。这个想法是探索数据管理的新范例，其中第三方服务提供商承载“数据库即服务”，为其客户提供无缝机制，以在主机站点创建、存储和访问其数据库。这种模式能够减轻组织很多负担，包括购买昂贵的硬件和软件，处理软件升级，聘请专业人员执行管理和维护任务等，这些工作都由服务提供商接管。Mehrotra 的团队已经在互联网上开发和部署了一个名为 NetDB2 的数据库服务，它在不断被使用。从某种意义上讲，NetDB2 支持的数据管理模式，为企业购买数据管理服务提供了一种有效的机制，从而使他们能够集中精力在核心业务上。另外一个备受关注的方向，是将 WSO2 数据服务中存在的 WS 概念，与 Mehrotra 及其团队正在进行的研究结合起来，将更高级的数据管理服务纳入标准。

7.2.8.3　软件即服务

另一个真正风靡的概念是软件即服务，也称为 SaaS（Software as a Service，软件即服务）。如本章参考文献［SOFT］所述，软件即服务是一种软件部署模式，提供商将应用程序许可给客户，作为服务按需使用。SaaS 软件提供商可以在自己的 Web 服务器上托管应用程序，或者将应用程序下载到消费者的设备，在使用后或在按需合同到期后禁用。按需功能也可以在内部处理，以在企业内部或由第三方（应用服务提供商，Application Service Provider，ASP）提供企业之间共享许可证。图 7.10 为 SaaS。

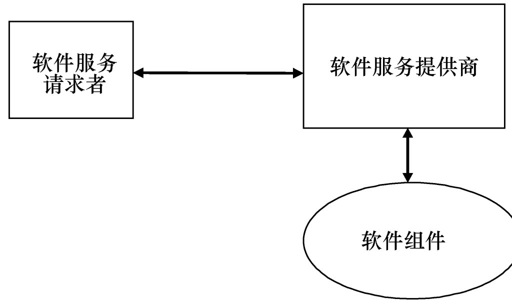


图 7.10　软件即服务

还要指出的是，SaaS 还可以利用面向服务的架构，来使软件应用程序相互通信。每个软件服务均可以充当服务提供商，通过公共代理将其功能告诉给其他应用程序；并且还可以充当服务请求者，使其他服务的数据和功能具体化。企业资源计划（Enterprise Resource Planning，ERP）软件提供商利用 SOA 来构建 SaaS 软件，例如 SAP AG 的 SAP Business ByDesign。

还有些人对 SaaS 持怀疑态度，因为他们声称这可能是相当昂贵的，并且有很多炒作成分。因此，一个公司必须做出权衡，是否要制作软件、购买软件，还是要将其作为 SaaS 模型的一部分进行授权等。在我们看到 SaaS 成为普遍趋势之前，仍然需要做很多工作。有关通过云计算提供这些服务的更多细节将在本书第三部分中讨论。

7.2.8.4 其他 X 即服务

还有其他几种类型的 X 作为服务。这些包括桌面即服务、网络即服务、平台即服务以及基础设施即服务。我们讨论两个这样的服务。如维基定义所述：“平台即服务（Platform as a Service，PaaS）是将计算平台和解决方案堆栈作为服务的交付。随着它的进一步发展，会提出一个软件开发平台，这个平台为了云堆栈顶层的云计算而设计。”这样的话，一个组织就可以调用服务，并获得硬件和软件堆栈并部署应用程序。在基础设施即服务（Infrastructure as a Service，IaaS）的情况下，维基定义说明如下：“基础设施即服务（IaaS）是将计算机基础设施（通常为平台虚拟化环境）作为服务的交付。这些虚拟基础设施堆栈是‘一切即服务’趋势的示例，并且共享许多常见特性。客户不是购买服务器、软件、数据中心空间或网络设备，而是将这些资源作为完全外包的服务购买。该服务通常在公用事业计算基础上计费，并且消耗的资源（因此产生的成本）将代表性地反映活动水平。这也是网络虚拟主机和虚拟专用服务器的演进。”

7.2.8.4.1 亚马逊 Web 服务

如亚马逊 Web 服务（Amazon Web Services，AWS）［AMAZON］的网页所述，自 2006 年初以来，AWS 已经为各种规模的公司提供了基于云的基础设施 WS 平台。使用 AWS，您可以请求计算能力、存储和其他服务。根据业务的需要，可以访问一套灵活 IT 基础设施服务。使用 AWS，您也可以灵活选择任何一个开发平台或编程模型，只要对您想要解决的问题最有意义即可。您只需支付您所使用的费用，无需支付前期费用或长期承诺，使 AWS 成为最具成本效益的方式是向您的顾客和客户提供应用程序。而且，通过 AWS，您可以利用 Amazon. com 的全球计算基础架构，这是 Amazon. com 零售业务的支柱。

AWS 有一些组件，包括数据库（称为 SimpleDB）、存储（Amazon S3）和云（Amazon EC2）。我们将描述本章参考文献［AMAZON］中讨论的一个这样的组件。其他组件的详细信息也在亚马逊网站中给出。Amazon SimpleDB 是提供数据索引和查询核心数据库功能的 Web 服务。该服务与亚马逊简单存储服务（Amazon Simple Storage Service，Amazon S3）和亚马逊弹性云计算（Amazon Elastic Compute Cloud，Amazon EC2）紧密结合，共同提供在云中存储、处理和查询数据集的能力。还要说明的是，Amazon WS 提供 SOAP 和 REST。

网格 WS 和云网格本质上是由用于执行各种应用程序的计算机集合组成。其目标是优化资源使用情况，并为各种任务安排机器。网格概念最近已经扩展到云中，其中存在虚拟计算空间，也包含映射到物理机器的大量虚拟机。这种概念被称为虚拟化。信息周刊［MART08］报道说，云计算代表了提供企业 IT 的一种新方式，在某些情况下是更好、更便宜的方式。通常，网格和云可以互换使用。一般而言，网格专注于调度资源，然而云重点是为企业提供一个有效的计算平台。

WS 在网格和云计算中发挥重要作用。Globus 联盟的目标是开发开放式网格服务架构（Open Grid Services Architecture，OGSA）。如本章参考文献［GLOB］所述，OGSA“代表了基于 Web 服务概念和技术的网格系统架构的演进。”Globus 联盟发布了一系列工具包，其中最新的是 Globus Toolkit 3.0 版本。它由“遵循 OGSA 原则的网格服务开源集合”组成。Globus 工具包还提供了一种开发环境，用于开发遵循 OGSA 原则的新网格服务。

云计算的 WS 包括上一节讨论的 AWS，还有 Google Apps 和 Salesforce.com CRM。这些云可以利用网格计算模式，如资源调度。可以期待的是，服务虚拟化将在云计算中发挥主要作用。在最近的一篇文章中，指出“服务虚拟化是一种从一个或多个预定义的服务文件创建虚拟服务的能力。服务文件通常是 Web 服务描述语言（WSDL）文件，通过服务容器业务应用程序产生，这些应用程序由 Java、.NET、PHP 类编程语言开发生成。”作者还指出，服务可能包括外包服务，如 SaaS、PaaS 或 IaaS 或内部服务。关于云计算技术的讨论，更多细节将会在第三部分给出。

7.3 专用 Web 服务安全

7.3.1 总述

在 7.3 节中，我们将论述各种类型的专用的 WS 安全。这将包括 WS 用于安全的数据、信息和知识管理以及领域 WS 的安全。这些 WS 使用第 5 章论述的 WS 的安全标准，例如 SOAP、XML、UDDI 和 WSDL 安全。

本节的结构如下。数据管理的 WS 安全将在第 7.3.2 节中论述。第 7.3.3 节将论述用于复杂数据管理（如地理空间数据管理）的 WS 安全。有关信息管理的 WS 安全将在 7.3.4 节中论述。有关知识管理的 WS 安全将在 7.3.5 节中论述。有关活动管理的 WS（如安全电子商务和 AIS）安全将在 7.3.6 节中论述。域 WS 安全将在 7.3.7 节中论述。一些新兴的 WS 安全技术将在 7.3.8 节中论述。有关数据和应用程序安全性的更多详细信息，请参阅本章参考文献［THUR05］。

7.3.2 用于数据管理的 Web 服务安全

各种安全数据库系统功能可以被作为 WS 调用。例如，查询管理器、事务管理和元数据管理器可以作为 WS 来执行。因此，要查询数据库，必须调用查询服务。该服务可

以调用查询翻译服务和查询优化服务。它还可以调用元数据服务来提取适当的元数据，也可以调用安全服务来检查访问控制策略。

接下来，可以使用语义 Web 技术来开发用于数据管理安全的语义 Web 服务。首先，安全策略可能用 XML 和 RDF 等语言表示，包括推理引擎在内的语义 Web 技术可能被应用于处理推理和隐私问题。例如，RDF 和 OWL 等语言可用于指定策略，然后可以凭借描述性的基于逻辑的引擎（例如 Pellet）来开发推理控制器，以确定通过推理的安全违规是否发生。



图 7.11 用于数据管理安全的 Web 服务

总之，在数据管理安全的各个方面，WS 以及语义 Web 技术都有应用。然而，数据管理技术和数据挖掘技术也可以应用于在网络上的管理和挖掘数据，以方便代理理解网页。图 7.11 为用于数据管理安全的 WS。

7.3.3 用于复杂数据管理安全的 Web 服务

7.3.3.1 地理空间数据管理安全

我们大部分的讨论都受到了与普渡大学的贝尔蒂诺教授和加利福尼亚大学戴维斯分校的格茨教授（见本章参考文献［BERT08］）合作研究的影响。这项研究也受到我们早期关于安全约束处理和保护多媒体数据的研究影响（见本章参考文献［THUR95］、［THUR90］）。Atluri 也对地理空间数据管理和安全（见本章参考文献［ATLU04］、［DAMI07］）进行了一些有意思的研究。

地理空间数据比关系数据更复杂。例如，我们可以对像素进行分类，并对构成地理空间数据的点和线进行分类。我们也可以根据内容、上下文和时间来定义策略。例如，拍摄的位置和图像组合起来可以归类，它们也可以解除归类，单独存在。此外，位置和图像可以在特定时间，或直到特定时间进行归类，然后可以解除分类。例如，伊拉克接管的卫星图像，可以从图像被捕获之日起六个月内进行归类，并在此之后取消分类。贝尔蒂诺和她的团队已经制定了地理空间数据的策略语言，和一个他们称之为基于角色的访问控制地理安全模型，该模型将 RBAC（Role - Based Access Control，基于角色的访问控制，RBAC）与地理空间数据进行了结合。

虽然明确需要执行地理空间数据的保密策略，但保证隐私仍然是一个挑战。保护地理空间数据的隐私意味着什么？今天，我们有能力进行监控，也可以在 Google Maps 中捕获图像。但是我们不能指望我们的房子的图像保持私密，因为房子的图像就在那里。不过，这是我的房子的事实是可以是隐私。我们需要研究地理空间数据的隐私管理问题。对于我们的语义 Web 研究，目标是开发地理空间 WS，它可以利用表示技术（如 GML 和 GRDF），以便我们获得机器可理解的网页。然而，为了确保地理空间语义 Web

安全，我们需要将地理空间语义 Web 技术与安全的地理空间数据管理技术相结合。这将是下一节的主题。图 7.12 说明了地理空间数据管理的安全性。

地理空间数据可以是流的形式。流数据的安全策略是什么？与地理空间信息系统密切相关的是运动着的数据库，数据不在一个地方。数据持续不断地变化，这些数据必须被捕获和管理。地理空间数据可能有各种来源，因此数据必须通过网络和其他方式进行整合。信息集成存在许多安全挑战。本章参考文献［BERT08］已经报道了这方面的一些工作。

我们正在制定一套地理空间信息系统策略，其数据表示将基于 OGC 开发的 GML。该策略将包括访问控制技术，例如基于角色和使用控制，以及信任、完整性、时间约束、数据货币、数据质量和数据来源等。之后，我们要设计和开发安全的地理空间 WS，展示与安全性相关的互操作性。我们将通过对地理空间信息系统推理，来检查安全违规。例如，已经开发了用于智能数据融合的推理技术。这种技术通过融合数据，可能会违反安全性限制。我们正在完善提出的各种技术，来处理关系数据的安全性，以达到处理融合图像的安全性。数据所有权和对盈利能力的需求，要求组织保护其信息库。当组织域由不只文本或数字等数据组成时，技术复杂度会增加几倍。一个地理空间 WS 的实例，它满足需要信息安全性的查询，其超出传统 RBAC 机制的水平。保护地理空间数据需要细粒度的访问权限，这使 RBAC 成为一个非最佳的解决方案。我们的目标是利用通过 OWL 提供的公理框架，来对潜在客户定义策略声明，并让推理机做内务处理。



图 7.12　地理空间数据管理安全

并不是地理空间机构拥有的所有数据都被自然而然认为是公共的。比如，这些数据可能包含有关暴露信息人的危险信息，这将危及他们的隐私。由于缺乏统一的安全框架，数据集成环境中的问题更加严重。如果数据集成的趋势持续发展，WS 提供商将很快执行复杂的服务，其需要将地理空间数据嵌入或组合到其他类型的数据中。然而，如果没有适当的安全架构在工作，数据交换中心会不情愿地、随意地提供数据。反过来，对数据的质量和有效性影响很大，因为客户只获得部分数据。

我们区分了在 WS 中最流行的两种安全性，并形成了语义 WS 架构的基础。第一类来处理 WS 用户的一般授权过程，以及超线程安全标准的任何后续执行。针对这种数据安全性的当前标准化协议集，包括加密方法、数字签名验证、证书生成和交换、WS 安全交换等。第二类涉及组织对数据的保护，这些数据可能来自入侵者或没有适当的访问权限的正当客户。在这方面使用最广泛的防御机制是各种形式的访问控制语言。我们正在开发一种语义丰富的、基于本体的访问控制解决方案，其用于地理空间数据，可以对全球地理空间数据集成的激增产生有利影响。

在应用层，地理空间数据的安全性可以划分为不同的逻辑段。我们的工作集中在客户或其他 WS 安全访问地理空间资源，尤其在动态组合的背景下。根据语义 Web 的愿景，我们正在用一种语言开发模块化的访问控制，使得推理认证执行引擎的发展成为可能。与基于 XML 的标准和基于逻辑的访问控制相反，我们对 OWL - DL 中的公理进行了定义，并强调了对策略的复用。根据以往以策略为中心的访问控制语言的实验，已经发现，对以个人为基础的资源定义策略，不太适合于集成的 GIS 应用。如果允许细粒度的资源访问，它将在决策或执行模块导航的策略文件中聚合策略，从而使整体查询的处理时间变得简短。我们的架构如图 7.13 所示。

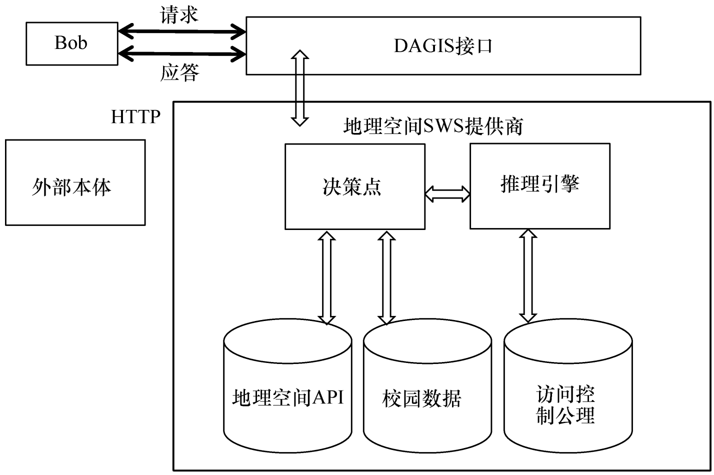


图 7.13　安全 DAGIS 架构

为了缩短策略的决策时间，我们的访问控制语言将所声明的规则集合作为一个独立的单元。然后，客户身份可以链接到适用于客户的规则。通过引用它们来模块化使用策略，其可以最大限度地减少规则重复。另一个重要的特征，是规则导航从策略向客户身份转移。我们框架中的地理空间语义 Web 服务代理接受已有身份或匿名的用户。

为了开发安全的地理空间语义 Web，我们需要在整个语义 Web 技术框架中整合安全性。OGC 等组织正研究在 GML 中制定的安全声明。此外，OASIS 等组织正在开发 GEO - XACML。我们正在开发在 GRDF 中制定的安全性声明，我们称之为 Secure GRDF。我们的重点是将安全的 DAGIS 框架扩展到安全的地理空间语义 Web 中。DAGIS 还具有基于 GRDF 的响应组件，并且可以推理安全策略。

7.3.3.2　安全的多媒体数据管理

安全的多媒体数据管理的功能包括安全查询处理和安全存储管理。这些功能可能是由 WS 提供支持。考虑到查询操作。查询 WS 将检查访问控制规则和安全约束，并相应地修改查询。例如，如果操作 X 存在被分类的事实，则该查询不能被发送到未分类的多媒体数据收集器，诸如视频摄像机之类，以拍摄事件。类似地，更新的处理器还检查访问控制规则，并计算要插入或修改的多媒体数据的级别。安全性也对多媒体编辑和浏览产生了影响。当浏览多媒体数据时，系统必须确保用户具有正确的访问权限，以浏览

链接或访问与链接相关联的数据。在多媒体编辑的情况下，当将不同级别的对象组合起来形成电影时，必须相应地对电影对象进行分类。可能需要对各种帧进行分类，或者分配高级水印，这种水印与构成影片的各个对象的级别相关联。此外，当编辑电影（例如删除电影的某些部分）时，需要重新计算编辑对象的级别。

接下来，考虑作为 Web 服务实现的存储管理器功能。存储服务必须控制对多媒体数据库的访问是安全的。存储管理器还负责根据安全级别划分数据。目前，对多媒体数据的访问方法和索引策略的安全影响尚未确定。而现在，为多媒体数据已经开发了许多索引策略，包括文本、图像、音频和视频。所以，我们需要检查策略并确定安全性的影响。另一个问题是多媒体数据的存储和显示之间的同步。例如，我们需要确保视频流畅显示，并且没有流量突发情况。这里可能存在恶意程序操纵存储和显示管理器，从而将信息从较高级别的进程隐蔽地传递到较低级别的进程。

7.3.4 信息管理的安全 Web 服务

如数据管理一样，诸如 XML，RDF 和 OWL 之类的语义 Web 技术可用于表示数据仓库、信息检索流和数字图书馆的安全策略，包括机密性、隐私性和信任策略。此外，基于例如 Pellet 的描述性逻辑的推理引擎，可以用于通过对数据仓库以及信息检索系统的推论，来推断未经授权的结论。语义 Web 技术也可用于表示非文本数据。例如，同步多媒体集成语言（Synchronized Multimedia Integration Language，SMIL）是视频的标记语言，而 VoiceML 是音频数据的标记语言。访问控制策略可以用例如 XML、RDF 或更具描述性的语言（如 REI）制定，他们可以在 SMIL 中表示的视频数据上强制执行。数据挖掘技术不仅可以应用于关系数据库，还可以应用于文本、语音、视频和音频数据库以及数字图书馆。使用数据挖掘，也存在隐私和安全问题。例如，数据挖掘可以推断敏感关联。因此，基于隐私保护的数据挖掘不仅在关系数据库上，而且在 XML、RDF 和 OWL 数据上仍然是一个挑战。

WS 在安全信息管理中有所应用，包括数据仓库安全、数据挖掘安全、信息检索安全和数字图书馆安全管理。例如，可以将数据仓库安全管理作为 Web 服务来调用。仓库安全管理员服务提供商将向目录注册其服务。请求仓库服务的用户将调用适当的服务。图 7.14 为用于信息管理的安全 WS 和语义 Web。



图 7.14　用于信息管理的安全 Web 服务

7.3.5 知识管理的安全 Web 服务

在前面的章节中，我们讨论了 WS 与知识管理之间的关系。也就是说，可以将知识管理服务作为 WS 提供，例如查找专家并共享演示文稿。安全的知识管理包括对知识管理实施适当的安全模型。例如，已经提出了用于知识管理

安全的各种访问控制模型和信息共享模型。这些安全服务可以作为安全 WS 提供。

语义 Web 技术为了知识管理在推理知识的同时，也可以推理安全策略，从而可以防止未经授权的推理。关于安全性，在 7.2.5 节沃伦所讨论的例子中，保密性、隐私权和信任策略将确定多大程度上 Sally 信任这些文献，并且可以使用这些文献来组织她的关于托尼·布莱尔的演讲报告。图 7.15 说明了知识管理安全与语义 Web 之间的关系。

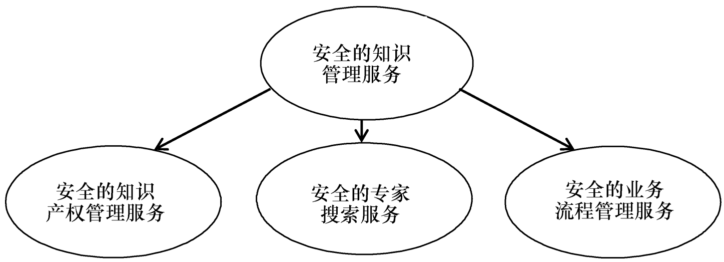


图 7.15　用于安全知识管理的 Web 服务

7.3.6　活动管理的安全 Web 服务

图 7.16 为用于活动管理的安全 WS。我们将在以下部分中分别讨论每个活动（例如，安全电子商务、安全协作和安全信息共享）。

7.3.6.1　安全电子商务

如前所述，电子商务是关于组织开展的业务交易，如销售商品和商业协议，以及消费者以电子方式从商家购买商品。电子商务方面有许多发展，其初步的进展在本章参考文献［THUR00］里讨论过。由于电子商务可能涉及企业之间数百万美元的商业交易，或者消费者和企业之间的信用卡购买交易，电子商务系统的安全是非常重要的。这种电子商务系统的示例包括电子支付系统和供应链管理系统。



图 7.16　用于活动管理的安全 Web 服务

Ghosh 和 Atallah 已经开展了一些关于安全电子商务以及安全供应链管理的工作（例如，本章参考文献［GHOS98］和［ATAL03］）。在电子支付系统的情况下，其挑战包括消费者和企业的识别和认证，以及追踪消费者的购买行为。例如，人们完全可以伪装成消费者，使用消费者的信用卡并以电子方式进行购买。因此，提出的一个解决方案是让消费者在进行某些购买时获得一些凭证。这些凭证可能是一组随机数，每次购买都会有所不同。这样一来，伪装成消费者的恶意进程可能没有对应凭证，因此可能无法进行购买。如果凭证也被盗，这将是一个问题。我们正在为安全的电子商务提出各种加密技术（参见本章参考文献［HASS00］）。也就是说，除了拥有凭证之外，信息可以用商家的公钥加密，并且只有商家才可以获得实际的数据。类似地，商家和消费者之间的通信也被加密。当商家之间

进行交易时，有关各方必须拥有一定的凭证，以便交易得到安全的执行。请注意，虽然电子商务交易和安全电子商务交易取得了很大进展，但将电子商务安全数据库交易管理技术融入到电子商务尚未成熟。本章参考文献［RAY00］已经报道了一些这方面的工作。

7.3.6.2 供应链管理安全

安全的供应链管理也是电子商务安全的重要方面。这里供应链指的是让组织为其他公司提供相关部件，以达到制造或其他用途。假设医院想从公司订购手术器械，那么医院和公司之间必须有一些谈判和协议。公司 X 可以从另一家公司 Y 请求部分零件，并且可能不想透露其正在为医院 A 制造部件的信息。这种敏感信息必须受到保护。供应链管理在许多领域的制造方面很有用，包括医疗、国防和情报。组织之间交流的一些信息可能非常敏感，特别是在军事和情报应用方面。因此，需要有一种保护这种敏感信息的方法。由于交易是在网络上执行的，所以正在提出访问控制规则和加密技术的组合，让其作为保护供应链管理敏感信息的解决方案。

自从 20 世纪 90 年代中期以来，我们一直在听到电子商务这个词，这是由于网络的爆炸式增长。虽然我们在开发信息技术方面取得了很大进展，例如数据库、数据挖掘和电子商务多媒体信息管理等，但安全性还有很多工作要做。此外，有关各种的个人信息也必须保密。我们在本书中讨论过的许多安全技术，包括安全的 Web 数据管理和安全的语义 Web，将适用于安全的电子商务。例如，语义 Web 可以用作执行电子商务功能的工具。通过开发机器可理解的网页，电子商务可以自动化，而不需要人的重复工作。这意味着语义 Web 的安全是至关重要的。随着我们在安全的网络信息管理技术方面取得进展，我们可以大大提高电子商务的安全性。电子商务应用程序可能被作为 WS 调用。

7.3.6.3 安全工作流和协作

如前所述，协作技术对于作为组织之间相互的电子商务来说是重要的。工作流是关于从开始到结束执行，诸如进行购买等操作的过程。这些步骤包括启动协议、转移资金并将货物发送给消费者。因为协作和工作流是许多业务的一部分，如电子商务和知识管理，所以我们需要安全的工作流和安全的协作。贝尔蒂诺等人已经做了很多工作。关于这个话题，最显著的发展是适用于安全的工作流管理系统的 BFA 模式（见本章参考文献［BERT99］）。最初在本章参考文献［DEMU93］中提出了一些关于安全协作系统的工作。此后，我们开发了几个方案（参见 IFIP 会议系列关于数据库安全部分）。在本节中，我们将进行安全工作流和协作的概述。

在安全的工作流管理系统中，这个想法是让用户拥有正确的凭证来执行特定的任务。例如，在进行项目购买时，只有项目负责人可以发起请求，秘书然后键入请求，之后管理员必须使用他/她的信用卡进行购买，邮箱也有权完成交付。本质上来说，我们所提出的是一个面向安全工作流的基于角色的访问控制模型。关于这个有了一些发展（见 SACMAT 会议记录），已经对安全工作流系统进行了各种技术的研究，如 Petri 网（见本章参考文献［HUAN98］）。

与安全工作流密切相关的是协作安全。协作的概念比工作流要广泛得多。工作流是为了执行任务，而必须进行的一系列串行或并行操作，协作是关于多个个体一起工作来解决问题的。一般的对象技术和特有的分布式对象管理技术正在被用于开发协作系统。在这里，环境中的个人和资源被建模为对象。个人和资源之间的通信被建模为对象之间的通信。通过对象请求代理进行通信。因此，为了安全的协作，针对对象请求代理商，要讨论一些安全问题。例如，所有相关各方是否应该获得访问资源的相同权限？如果访问资源权限不同，那么个人间如何共同协作共享数据呢？

工作流和协作是关于组织或团体共同工作努力实现目标，如设计系统或解决问题。协作技术对于作为组织之间相互的电子商务来说是重要的。

信任和协商系统也在工作流和协作系统中发挥重要作用。例如，各方如何相互信任来解决问题？如果 A 给 B 一些信息，且 A 和 C 不相互通信，B 可以与 C 共享信息吗？当我们讨论安全联盟时，也提出了类似的问题。此外，数据管理安全技术对于管理工作流和协作应用程序的数据是必要的。虽然在这方面，取得了很大进展，但还是有很多事情要做，特别是对于语义 Web 和新兴技术的发展，如对等数据管理。

在第 5 章中，我们讨论了复杂活动的 WS，如工作流程、编排和编配。此类活动的安全性处于起步阶段。例如，调用组成工作流程应用程序的 WS 中的安全问题是什么？编排和编配的安全问题是什么？WS 组合的安全问题是什么？

7.3.6.3.1　信息互操作性安全

在整合信息方面有几个难题，特别是在异构环境中。一个是架构异质性，其中系统 A 基于关系系统，而系统 B 基于对象系统。也就是说，当两个系统基于不同的模型时，我们需要解决这个问题。一种选项是拥有一个共同的数据模型。这意味着两个系统的构造必须被转换成公共数据模型的构造。当我们考虑安全属性时，我们必须维护各个系统，确保策略得到执行。为了联合数据库系统的安全，多个模式被集成，以形成联合模式。实际上，我们采用了谢恩和拉森的架构来实现安全的联合环境。本章参考文献［THUR94］中讨论了集成异构模式时的一些安全挑战。我们假设每个组件将一个模式导出到联盟中。然后，这些模式被集成以形成联合策略。在安全的环境中，我们需要确保在整个联盟中维护各个系统的安全属性。在下一节中，我们将讨论安全策略集成问题。

下面，我们将重点关注策略整合。本章参考文献［THUR94］报道了对于联合数据库的安全策略整合的初步调查。在此，我们假设异构数据库系统必须被集成，以形成安全的联合数据库系统。对于模式集成，我们的方法非常类似于谢恩和拉森（见本章参考文献［SHET90］）。在策略集成的情况下，每个系统将安全策略导出到联合体。我们假设组件系统对外部用户有更严格的访问控制要求。也就是说，导出策略除了本地系统执行的规则之外，还可能具有访问控制规则。这里的挑战是要确保在联合体层面没有安全违规。

当实体在不同的位置被解释不同，或者不同的实体被解释为相同的对象时，会发生语义异质性。例如，术语“速度”可以在节点 1 中以 mile/h 为单位，在节点 2 中可以是 km /天。另一个例子是约翰·史密斯，在节点 1，他是约翰·史密斯（John Smith），

在节点 2，他却是约翰·K·史密斯（John K Smith）。在这两种情况下，相同的实体但解释是不同的。另一方面，节点 1 的约翰·史密斯可以是真的约翰·J·史密斯，在节点 2 他是约翰·K·史密斯。他们是不同的人，但却错误地认为是相同的。

语义异质性是数据集成和信息互操作性的主要挑战之一。它们不仅发生在关系数据库中，还发生在对象数据库、多媒体数据库甚至地理空间数据库中。例如，当异构地理空间数据库被集成时，每个数据库可以用不同的方法来表示相同的坐标系。自 20 世纪 90 年代以来，已经提出了用于处理异构性的各种解决方案，然而只是最近我们使用语义 Web 技术，才对该问题有了很好的处理。

可以调用 WS 来集成异构数据库。也就是说，代表用户的代理可以调用 Web 服务进行集成。该 Web 服务可以利用本体来处理语义异质性来执行操作。因此，为了拥有安全的信息互操作性，我们需要一个安全的 WS。对 WS 的信息互操作性安全的研究刚刚开始，这些研究包括具有隐私保护的本体一致性和 WS 的安全策略集成。图 7.17 说明了 WS 的信息互操作性安全包括的各个方面。

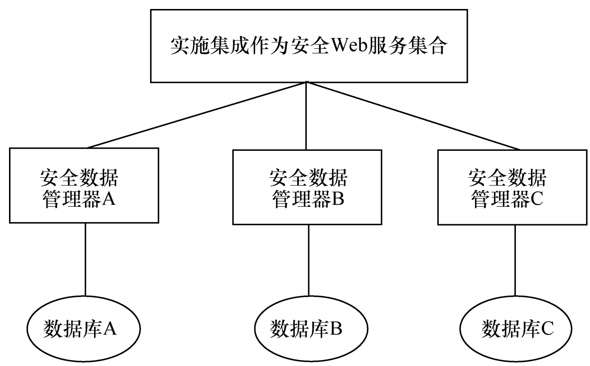


图 7.17　用于安全信息互操作性的 Web 服务

7.3.6.3.2　确保（安全的）信息共享

确保（安全的）信息共享（AIS）是关于组织之间共享信息，但同时执行策略和程序，以便对数据进行集成和挖掘来提取有用信息。例如，必须整合来自多个安全级别的各种数据源，以及来自不同的服务机构（包括空军、海军、陆军、地方、州和联邦机构）的数据，以便挖掘数据，提取模式和信息，确定关系和做出决定。数据库包括许多种类，例如含有有关军事战略信息的军事数据库、含有潜在恐怖分子信息的情报数据库及其攻击模式，以及含有相关传染病和累积信息的医学数据库。数据可以是结构化的或非结构化的，包括地理空间/多媒体数据。数据还需要在医疗机构之间共享，如医生、医院和药店。除非数据整合并形成一张大图，否则，向所有相关方面通知发生的事件将会很困难。虽然不同的机构必须共享数据和信息，但他们还需要执行适当的安全性和完整性策略，以便数据不会被未经授权的个人掌握。实际上，代理间必须共享信息，同时要保持安全性和完整性。联盟由一系列组织组成，可以是在对等环境中共同努力解决诸如情报和军事行动以及医疗保健行动等问题的机构、大学和企业。联盟通常是动态的。也就是说，会员可以按照政策和程序加入联盟。确保联盟的安全运作是一项挑战。我们认为，联盟的成员也称其为合作伙伴，可能是值得信赖的、不可信的或部分（半）值得信赖的。马克尔报告（见本章参考文献［MARK03］）中讨论了联盟数据共享的各个方面。

与此同时，我们迫切需要多个组织共享数据，同时执行安全策略。这些策略包括保密、隐私和信任策略。例如，患者数据可能由多个组织共享，包括医院、各级政府和机构。保护患者数据的隐私很重要。然而，同样重要的是，要去掉不必要的访问控制，其会导致禁止信息的共享。人们需要灵活的策略，以便在紧急情况下，所有的数据都是共享的，这可以让人们做出有效的决定。在正常运作中，保持机密和隐私是重要的。此外，信任策略确保数据在受信任的个人之间共享。在该领域的标准包括 RBAC（见本章参考文献［SAND96］）以及 P3P（隐私偏好保护平台，Platform for Privacy Preferences）。

存在两种类型的矛盾：一种是安全性与数据共享。数据共享的目标是为组织尽可能多地共享数据，以便数据被挖掘并获得关键信息。但是，当执行安全策略时，并不是所有的数据都被共享。另一种矛盾的类型是在实时处理和安全之间。作战人员需在恰当的时间得到信息。即使哪怕晚 5min，信息可能都没有用了。这意味着如果要执行各种安全检查，那么信息可能无法及时到达作战人员手中。

WS 在信息共享中发挥重要作用。例如，组织 A 可以调用 Web 服务以从另一个组织 B 中获取信息。该 Web 服务可以调用另一个 Web 服务，来确定组织 B 共享信息的诱因是什么。在 AIS 的情况下，组织 B 还可以检查安全策略，以确定信息是否可以共享。我们正在建立 AIS 生命周期，面向服务的架构是我们方法的核心内容，细节将在附录 D 中讨论。

7.3.6.3.3　社交网络安全

社交网络近年来爆炸式增长。我们现在有 Facebook、Friendster 和 Twitter 等。目前社交网络数据正在被挖掘，从而提取有用的信息，和向会员提供更好的服务。此外，采集的信息也可用于帮助反恐和执法。与此同时，保护合法公民的隐私也十分重要。

最近人们对保护社交网络以及社交网络开发隐私保护技术方面很感兴趣。例如，会员如何在隐私得到确保的前提下，透露正确的信息量？系统是否会向会员提供一些反馈，说明其透露了太多的信息？也许可以调用 WS 来管理社交网络。例如，会员可以调用 Web 服务在网络上发布信息。当然，这些 WS 必须确保适当的安全和隐私策略。

7.3.6.3.4　供应链管理安全

供应链管理和物流的安全性受到很大的关注。在物流方面，一个主要目标是物体的安全移动。当物品必须从位置 A 移动到 B 时，物品必须在物理上和数字上都被保护。RFID 技术正在用于跟踪物流和供应链管理的对象。因此，安全的 RFID 技术至关重要。

在供应链方面，还需考虑其他的安全。例如，可以使用几个组件来制造产品。这些组件可能来自世界各地，可能会有安全影响。即使组件可能受到威胁，但仍然需要保证产品是安全的。这是一个非常具有挑战性的问题。如 7.2 节所述，可以调用 WS 来执行供应链管理以及物流业务。这些 WS 必须是安全的，以确保供应链管理安全。

7.3.7　安全的领域 Web 服务

7.3.7.1　防御

国防部正在使用服务技术和基于面向服务的架构模式的全球信息网格。但其许多发展受到 XML 和本体的影响。正如我们所说，WS 也在国土安全应用中发挥着重要作用。

对于许多领域的应用程序，重点是实施基于 XACML 安全标准的带有属性的访问控制。其目标是让用户展示他/她的凭证并请求资源。然后，基于被执行的策略和用户的凭证，系统将决定用户是否可以访问资源。这实际上是策略决策点的功能。随后，策略执行点将强制执行访问请求。

7.3.7.2 医疗保健和生命周期

如前所述，语义 Web 技术在医疗保健信息技术的发展中发挥了重要作用。例如，我们已经为电子医疗保健记录以及生命科学中的几个项目开发了本体。医疗信息系统的主要安全挑战，是涉及隐私问题，其目标是确保患者信息记录的隐私性。通常，患者要确定在什么条件下要保护哪些信息。当一个组织需要有关患者的数据时，要制定相关策略。如果策略与病人所指定的一致，那么这些信息就会发布给组织。可以使用诸如 P3P 之类的 W3C 标准来确保隐私安全。此外，整个操作可以作为 WS 来实现。

7.3.7.3 金融

由于金融领域涉及金钱，因此机密性对于金融数据至关重要。尽管由于患者隐私的缘故，从而难以获得医疗保健数据，但我们发现，几乎更不可能获得如信用卡数据这样的财务数据去进行研究。几个小组正在为金融领域开发 WS 和语义 Web 技术。正如我们所说的，马德里的小组对应用语义“基于本体的平台”做了一些非常好的研究，该平台提供①在知识库中集成内容和语义，其提供了对低级内容的概念性视图，②基于超媒体的自适应知识可视化和导航系统，③语义搜索设施“（见本章参考文献［CAST］）。我们面临的挑战是将客户信息集成到这些语义 Web 技术中去的同时，也要集成安全和隐私策略去保护金融数据。然后，才是 WS 执行金融操作。

7.3.7.4 其他领域

电信领域的安全性也得到了广泛的研究。例如，正在提出用于移动应用的 WS（包括手机和平板电脑）。此外，电信和移动计算领域的安全措施正在爆炸式发展。这些领域的安全 WS 也在研究之中。图 7.18 为各种安全领域中应用程序的 WS。

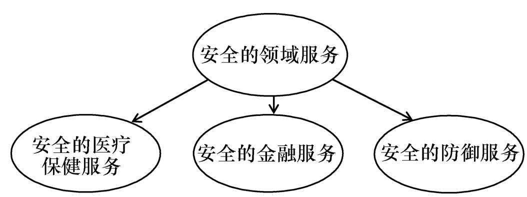


图 7.18 安全的领域 Web 服务

7.3.8 新兴 Web 技术的安全

7.3.8.1 X 即服务的安全性

对于 X 即服务，正在变得非常受欢迎。其中 X 可能是数据、软件或其他一些概念，例如平台、操作系统、编译器或基础设施等。在服务计算领域，一切都将成为一项服务，包括数据以及医疗保健和金融等现实服务。

近年来数据服务的安全性得到了研究。随着外包工作的爆炸式增长，许多面向数据的工作正在被外包。因此，保护数据的敏感性方面至关重要。此外，当数据被用作服务提供商的服务时，重要的是数据的质量高且不被破坏。关于软件，重要的是作为服务使用的软件是无错误的，且不被蠕虫病毒感染。图 7.19 说明了为 X 即服务提供安全性的概念。关于这个主题的更多细节将在本书中其他部分讨论。

7.3.8.2　亚马逊 Web 服务的安全性

AWS（Amazon Web Services）已经为各种规模的公司提供了云的基础设施 WS 平台。使用 AWS，可以申请计算能力、存储和其他服务。AWS 提供名为 SimpleDB 的数据库组件，它提供数据库功能，如查询。亚马逊使用 SOAP 来进行客户端和服务提供商之间的通信。因此，与 SOAP 相关的所有安全问题都是相关的。此外，云计算安全问题也与这些讨论相关。在云计算环境下，安全问题包括实施适当的访问控制策略，以及数据在多个位置的安全存储。最近，研究人员正在探索与云中的虚拟化有关的审查问题。有关云计算安全问题的更多细节将在第五部分讨论。

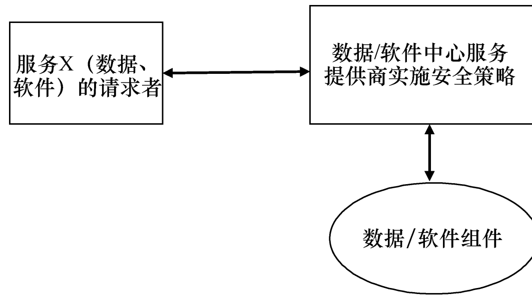


图 7.19　X 即服务的安全性

我们的研究重点是对 AWS 实施的访问控制和加密。要对存储在亚马逊环境中的数据进行加密，从而保护它们。此外，我们还在这种环境中实施基于角色的访问控制。还有，我们在云环境中实施了 XACML，下一节将详细介绍它。有关此主题的更多详细信息，请参见本章参考文献［PRAN09］。

7.3.8.3　为云和网格提供安全的 Web 服务

我们迫切需要安全地存储、管理、共享和分析大量复杂（例如半结构化和非结构化）的数据，以确定模式和趋势，从而提高医疗质量，更好地维护国家安全和探索可替代能源。新兴云计算模式试图解决网络连接设备的发展问题，并处理大量的数据。Google 现在已经引进了 Map/Reduce 框架来处理商品硬件上的大量数据。Apache 的 HDFS 正在成为云计算的高级软件组件，它还结合了 Map/Reduce 等集成部件。增加人类推理、解释和决策能力的需求导致了语义 Web 的出现，这是一个尝试将网络从目前的只有人类可读的形式转变为机器可处理形式的一个举措。反之也产生了许多拥有大量数据的社交网站，以便共享和管理。

我们正在云计算和云计算安全方面进行广泛的研究（见本章参考文献［HAML10］）。关于这方面的大部分讨论都是基于这一研究。图 7.20 为基于 Hadoop 和 Map/Reduce 的安全云。

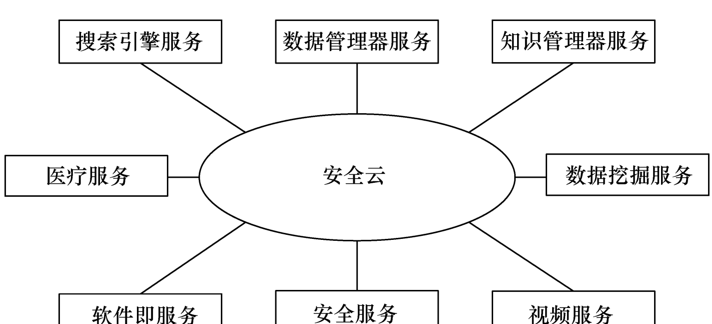


图 7.20　安全云

7.4 总结和展望

在本章中，我们讨论了各种类型的专用 WS。首先，讨论了数据管理和复杂数据管理的 WS。然后，讨论了信息管理和知识管理的 WS。其次，讨论了 WS 的活动管理和特定领域的 WS。再次，我们讨论了一些新兴的 WS，包括“X 即服务”的范例。最后，我们描述了新兴 WS 的安全性，包括 WS 的数据、信息和知识的安全性以及活动管理。我们还讨论了特定领域的 WS 以及 AWS 和云的安全性。我们还检查了 X 即服务的安全性。

我们相信，今后关于 WS 以及 WS 安全大部分的进一步研究将以本章讨论的主题为基础。有部分原因，是因为 WS 和 WS 安全是云计算和云计算安全的基本内容。由于需要处理和分析大量的数据，云计算技术正在迅猛发展。本书的其余部分将致力于云计算和云计算安全的论述。

参考文献

[ALAM06] Alam, A. and B. Thuraisingham, *GRDF and Security Constructs*, *Computer Standards and Interfaces*, 33(1), January 2011, 35–41.

[AMAZON] http://aws.amazon.com/.

[ATAL03] Atallah, M., H. Elmongui, V. Deshpande, and L. Schwarz, Secure supply-chain protocols. *IEEE International Conference on Electronic Commerce* (CEC 2003). Newport each, CA.

[ATLU04] Atluri V. et al., An authorization model for geospatial data. *IEEE Transactions on Dependable and Secure Computing*, 238–254, December 2004.

[BERT99] Bertino, E. et al., The specification and enforcement of authorization constraints in workflow management systems. *ACM Transactions on Information and Systems Security*, 2(1), 65–104, 1999.

[BERT08] Bertino, E., B. Thuraisingham, M. Gertz, and M. Damiani, Security and privacy for geospatial data: Concepts and research directions. *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS (SPRINGL 2008):* 6–19, Vreine, CA.

[BORD] Borden, J., *XML Healthcare: From ASTM XML DTDs to the Semantic Web*, Technical Report, Tufts University.

[CAST] Castells, P., B. Foncillas, R. Lara, M. Rico, and J.L. Alonso, Semantic web technologies for economic and financial information management. http://nets.ii.uam.es/aniceto/publications/esws04.pdf.

[CHEN04] Chen-Burger, Y.-H., K.-Y. Hui, A.D. Preece, P.M.D. Gray, and A. Tate, Supporting collaboration through semantic-based workflow and constraint solving. http://stoater.inf.ed.ac.uk/project/ix/documents/2004/2004-ekaw-chenburger-kraft-ix.pdf.

[DAMI07] Damiani, M., E. Bertino, B. Catania, and P. Perlasca. GEO-RBAC: A spatially aware RBAC. *ACM Transactions on Information and System Security*, 10(1), 1–42, 2007.

[DATA] XML.com: http://www.xml.com/pub/a/2007/10/25/data-sources-as-web-services.html.

[DEMU93] Demurjian, S. et al., Security for collaborative computer systems, *Multimedia Review: The Journal of Multimedia Computing*, Penton Media Publishers, 4(2), 40–47, Summer 1993.

[EBXM] EBXML.org: http://www.ebxml.org/.

[GHOS98] Ghosh, A., *E-commerce Security, Weak Links and Strong Defenses.* John Wiley, New York, NY, 1998.

[GLOB] http://en.wikipedia.org/wiki/Globus_Alliance.

[HAML10] Hamlen, K. et al., *Security Issues for Cloud Computing*, Technical Report, University of Texas at Dallas, January 2010.

[HASS00] Hassler, V., *Security Fundamentals for E-Commerce*, Artech, London, UK, 2000.

[HUAN98] Huang, W. and V. Atluri, Analyzing the safety of workflow authorization models, In *Proceedings of the IFIP Database Security Conference*, Chalidiki, Greece, July 1998 (formal proceedings published by Kluwer, 1999).

[LIFE] Ontologies for the life sciences. www.jonathanborden-md.com/HealthcareSemWeb.ppt.

[KREM] Schulze-Kremer, S., Ontologies for the life sciences, Encylopedia of Genetics, Genomics, Proteomics and Bioinformatics, 2005. Universtät Hannover, Hanover, Germany.

[MARK03] Markle Report on Assured Information Sharing, 2003.

[MART08] InformationWeek, June 2008, Guide To Cloud Computing Richard Martin and J. Nicholas Hoover, http://www.informationweek.com/services/hosted-applications/guide-to-cloud-computing/208700713

[OBEL] E3VALUE.com: http://www.e3value.com/projects/ourprojects/obelix.

[OGC] OGC White Paper, *An Introduction to GeoRSS: A Standards Based Approach for Geo-Enabling RSS Feeds*, http://www.opengeospatial.org/pressroom/papers.

[PALI04] Paliwal, A.V., N. Adam, C. Bornhövd, and J. Schaper, Semantic discovery and composition of web services for RFID applications in border control. http://www.ai.sri.com/SWS2004/final-versions/SWS2004-Paliwal-Final.pdf.

[PRAN09] Pranav, P., *Security for Amazon Web Services*, MS Thesis, University of Texas at Dallas, November 2009.

[RAY00] Ray I. et al., A fair exchange e-commerce protocol with automated dispute resolution. In *Proceedings of the IFIP Database Security Conference*, Amsterdam, The Netherlands, August 2000.

[SAND96] Sandhu, R., E. Coyne, H. Feinstein, and C. Youman, Role-based access control models. *IEEE Computer*, 29(2), 38–47, February 1996.

[SHET90] Shet, A. and J. Larson, *Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases*, Bellcore, Technical Report TM-STS-016302, Piscataway, NY, 1990 (also published in *ACM Computing Survey*, 9/1990).

[SING06] Singh, R. and Salam, A., Semantic information assurance for secure distributed knowledge management: A business process perspective. *IEEE Transactions Systems Man and Cybernetics*, 36(3), 472–486, 2006.

[SOFT] http://en.wikipedia.org/wiki/Software_as_a_service.

[STUD] Semantically enabled knowledge technology, http://semanticweb.org/wiki/SEKT.

[STUD] http://www.emeraldinsight.com/journals.htm?issn=0143-5124&volume=26&issue=4/5&articleid=1501536&show=html

[TELE] http://en.wikipedia.org/wiki/Parlay_X.

[THUR90] Thuraisingham, B., Multi-level Security for multimedia database systems. *IFIP Data Security Conference*, 99–116, 1990. Halifax, UK.

[THUR94] Thuraisingham, B., Security issues in federated database management systems.

Computer and Security, 13(6): 509–525, 1994.
[THUR95] Thuraisingham, B. and W. Ford, Security constraint processing in a multilevel distributed database management system. *IEEE Transactions on Knowledge and Data Engineering*, 7(2), 278–293, April 1995.
[THUR97] Thuraisingham, B., *Data Management Systems Evolution and Interoperation*, CRC Press, Boca Raton, FL, 1997.
[THUR00] Thuraisingham, B., *Web Data Management and Electronic Commerce*, CRC Press, Boca Raton, FL, 2000.
[THUR05] Thuraisingham, B., *Database and Applications Security*, CRC Press, Boca Raton, FL, 2005.
[THUR07] Thuraisingham, B. et al., *DAGIS Architecture*, Technical Report, University of Texas at Dallas, 2007.
[THUR10] Thuraisingham, B., *Secure Semantic Service Oriented Systems*, CRC Press, Boca Raton, FL, 2010.
[TRAS] Trastour, D. et al., Semantic web support for the business-to-business e-commerce lifecycle. http://www.hpl.hp.com/techreports/2002/HPL-2002-3R1.pdf.
[WARR06] Warren, P., Knowledge management and the semantic Web: From scenario to technology. *IEEE Intelligent Systems*, 21(1): 53–59, 2006.
[WFMC] Workflow management coalition, http://xml.coverpages.org/wf-xml.html.
[XML] Thinking XML: A glimpse into XML in the financial services industry. http://www-128.ibm.com/developerworks/xml/library/x-think22.html.
[ZHAO] Zhao, G. et al., Engineering an ontology for financial securities fraud, http://www.starlab.vub.ac.be/research/projects/poirot/Publications/eofsf_20-08-final.pdf.

第二部分总结

第二部分概述了 Web 服务和安全的 Web 服务。因为云给消费者提供了一系列服务，所以 Web 服务是云计算的核心。

第 5 章概述了面向服务的计算模式和服务的安全问题。首先，我们论述了服务、SOA 和 Web 服务的概念，以及新兴的 X 即服务的范例、SOA 和设计。然后，我们论述了 SOA 和 Web 服务的安全性。特别是，我们论述了 Web 服务的访问控制，诸如 SAML 和 XACML 的标准，以及一些新兴的安全模型，如授权、信息流和多层次安全性、模型。我们还论述了 Web 服务的身份管理。

第 6 章概述了语义 Web 技术和语义 Web 服务的概念。特别是，我们论述了 Tim Berners Lee 的语义 Web 技术，以及 XML、RDF、本体和 Web 规则。我们还论述了语义 Web 服务，以及如何利用语义 Web 技术。如第 6 章所述，语义 Web 服务是语义云计算的核心。

第 7 章论述了各种类型的专用 Web 服务。首先，我们论述了数据管理和复杂数据管理的 Web 服务。其次，我们论述了信息管理和知识管理以及活动管理的 Web 服务。之后是对特定领域的 Web 服务的论述。最后，我们论述了一些新兴的 Web 服务，包括“X 即服务”的范例。

第三部分　云计算概念

第三部分简介

目前，我们已经给出了 Web 服务的概念，现在准备介绍云计算，概述有关云计算的各种概念、功能、技术、产品以及相关标准。我们还将说明如何把第二部分讨论的关于 Web 服务、语义 Web 服务和专用 Web 服务的概念应用于提供云服务的过程中。

第三部分包含 5 章：第 8、9、10、11 和 12 章。第 8 章对云计算进行了概述。第 9 章则论述云计算的一些功能。第 10 章论述云数据管理，这是我们研究与开发的重点。一些专用的云服务（比如云应用中的移动云）将是第 11 章所要讨论的内容。最后第 12 章的主要内容是云计算服务的提供商、产品和框架结构。

第 8 章　云计算概念

8.1　概述

新兴的云计算模型试图致力于解决网络连接设备增加的问题，并处理大量的数据。谷歌现在已经推出了用于处理常用硬件中庞大数据的 Map/Reduce 框架。Apache 的 HDFS 正在成为云计算的高级软件组件，并结合了集成部件，如 Map/Reduce（见本章参考文献［HDFS］）。包括惠普的开放云测试平台在内的一些云正在使用 HDFS。这反过来导致了拥有庞大数据的许多社交网站被分享和管理。比如，我们可能想要从统计学上分析股市历史数据来揭示一种模型，或者根据前几年天气的相关数据建立一个稳定的天气模型。为了处理这么多来自不同网站（即节点）的数据，我们需要可扩展的硬件和软件组件。已经出现的云计算模型就是为解决网络连接设备爆炸式增长的问题，并能处理大量数据。它具有大规模的可扩展性和新互联网驱动的经济等定义和特征。

在本章中我们将论述一些云计算的初步知识。在这一部分其他章节还将论述一些技术，比如 Hadoop 和 Map/Reduce。我们会首先介绍一下云计算的意义。虽然对于云计算已经给出了许多定义，但我们将采用 NIST 给出的定义。这是一个基于服务的云计算定义。特别的是，我们将详细阐述一些在第 5 章介绍过的内容（即 Web 服务）。接下来我们将论述包括虚拟化和数据存储在内的各种核心概念。

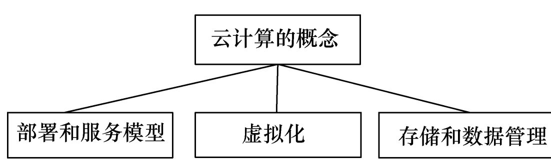


图 8.1　云计算概念

本章的结构如下：8.2 节讨论云计算初步知识；虚拟化将在 8.3 节中讨论；8.4 节讨论云存储和数据管理问题；本章总结在 8.5 节。图 8.1 为本章中所述的组件。我们还将指出第二部分讨论的概念对第三部分讨论的概念的适用性。

8.2　云计算初步知识

如本章参考文献［CLOUD］里所述，传统计算以产品的形式提供，而云计算将计算变为一种可提供的服务。因此，用户根据即付即用模式支付服务费用。云提供的服务可能包括硬件、系统、数据和存储。云的用户不需要知道软件和数据的位置，也就是说，云提供的软件和数据服务对用户来说是透明的。本章参考文献［NIST］将云计算定义如下：

云计算是一个模型，能实现无处不在的、方便的、按需的通过网络访问可配置的计算资源共享池（如网络、服务器、存储、应用程序和服务），且只需最少的管理工作或与服务提供商的互动，就可以快速配置和发布。

云模型由多个部署模型和服务模型组成。下面介绍这些模型。

8.2.1 云的部署模型

云计算有多种部署模型，这包括公有云、社区云、私有云和混合云。对于公有云，服务提供商通常提供可由公众访问的 WWW 上的服务。这种云可以提供免费服务或者是即付即用的服务。社区云是一群组织一起开发的云。这些组织往往有着共同的目标，比如提供安全和容错性功能，成本由参与其中的组织共同承担。此外，社区云可以由组织或第三方托管。私有云是专门针对特定的组织而开发的云基础架构。这也可以由组织或第三方托管。混合云由公有云和私有云组合而成。这样在混合云中，组织既可以使用私有云进行高度敏感的服务，同时又可以使用公有云来进行较低敏感度的服务，还能充分利用 WWW 提供的功能。Kantarcioglu 和他的同事们表示，混合云将是未来的部署模型（见本章参考文献［KHAD12］）。图 8.2 为云的部署模型。

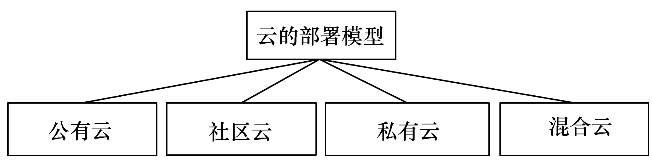


图 8.2　云的部署模型

8.2.2 服务模型

如前所述，云计算提供了各种服务。这些包括基础设施即服务（Infrastructure as a Service，IaaS）、平台即服务（Platform as a Service，PaaS）、软件即服务（Software as a Service，SaaS）和数据即服务（Data as a Service，DaaS）。在 IaaS 中，云提供了一系列硬件和网络，供一般公众或组织使用。用户安装操作系统和软件来运行应用程序。用户将根据他们用于计算的资源进行结算。在 PaaS 中，云提供商将向用户提供系统软件，如操作系统（OS）和执行环境。用户将加载他们的应用程序，并在云提供的硬件和软件基础设施上运行。在 SaaS 中，云提供商将为用户提供运行的应用程序。这些应用程序可以说是计费应用程序、税务计算应用程序和销售工具。云用户通过云客户端访问应用程序。在 DaaS 的情况下，云提供数据给云用户。数据可能存储在云用户要访问的数据中心里。需要注意的是，虽然 DaaS 用于表示桌面即服务，但最近它表示数据即服务。图 8.3 为云的服务模型。应注意到第 5 章中讨论的所有面向服务的概念、技术和标准都可以用于开发基础设施、平台、软件、应用程序和数据服务等云服务。

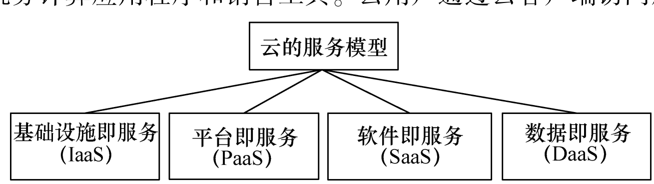


图 8.3　云的服务模型

8.3 虚拟化

虚拟化实质上意味着创造某种不存在的不是实际的东西。它可以是硬件、软件、内存和数据。虚拟化的概念在计算方面已存在数十年。早在20世纪60年代，就引入了虚拟内存的概念。这个虚拟内存给应用程序带来了它有连续的一块工作内存的错觉，映射是为了将虚拟内存映射到实际物理内存而开发的。

硬件虚拟化是云计算的基本概念。实际上，是创建一个托管在具有操作系统的真实计算机上的虚拟机。这意味着当实际的机器可能是运行Windows操作系统的IBM PC（个人计算机）时，通过虚拟化，它可以向用户提供运行Linux的SUN Solaris机器。实际的机器称为主机，而虚拟机称为客户机。

其他类型的虚拟化包括操作系统级虚拟化、存储虚拟化和数据/数据库虚拟化。操作系统级虚拟化与硬件虚拟化密切相关。在这种类型的虚拟化中，多个虚拟环境可被创建在单个操作系统中。虚拟机监视器（也称为管理程序）是虚拟机运行在主机上的软件。在存储虚拟化中，逻辑存储从物理存储中抽象出来。这必须从逻辑存储提供映射到物理存储中。在数据/数据库虚拟化中，数据从基础数据库中抽象出来。这样，用户就有像在自己的数据库上工作的错觉。我们可以创建多个这样的虚拟数据库。虚拟数据库必须映射到物理数据库。应该注意的是，尽管有些人区分了数据虚拟化和数据库虚拟化，但是我们仍可以互换使用这两个术语。在网络虚拟化中，虚拟网络被创建。虚拟网络必须映射到物理网络。图8.4为各种类型的虚拟化。

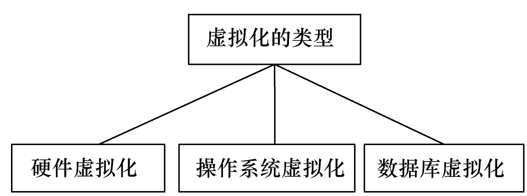


图8.4 虚拟化的类型

如前所述，云计算的核心是虚拟机管理程序或虚拟机监视器的概念。硬件虚拟化技术允许多个操作系统（称为访客）在主机上同时运行。这些多个操作系统共享虚拟化的硬件资源。虚拟机管理程序不是一个新词，在20世纪60年代中期，IBM 360/65机器中就首次使用过。现有不同类型的虚拟机管理程序。其中有一种模型，管理程序在主机硬件上运行，并管理客户操作系统。现在的流行虚拟机VMware和XEN均基于此模型。在另外一个模型中，管理程序在常规操作系统环境中运行。虚拟机也被并入到嵌入式系统和手机中。嵌入式管理程序具有实时处理能力。我们将在第9章中提供有关虚拟机管理程序的更多详细信息，如XEN和VMware。虚拟化的一些细节在本章参考文献［VIRTUAL］中提供。

8.4 云存储和数据管理

在云存储模型中，服务提供商为数据中心的客户存储大量数据。将从作为托管公司

的服务提供商那里，那些需要存储空间的人可以租用存储空间。这种情况下，数据的实际位置对用户来说是透明的。向用户呈现的是虚拟化存储，存储管理器将虚拟存储映射到实际存储，并为客户管理数据资源。单个对象（例如，客户的整个视频数据库）可以存储在多个位置。每个位置可以存储多个客户的对象。图 8.5 为云存储管理。

可视化云存储有很多优点。用户不需要购买昂贵的存储设备，数据可以放在云中的任何地方。云环境提供像备份和恢复等维护工作，目的是让用户快速访问云端。然而，由于数据的所有者不能完全控制他的数据，所以将数据存储在云中时可能存在严重的安全问题。

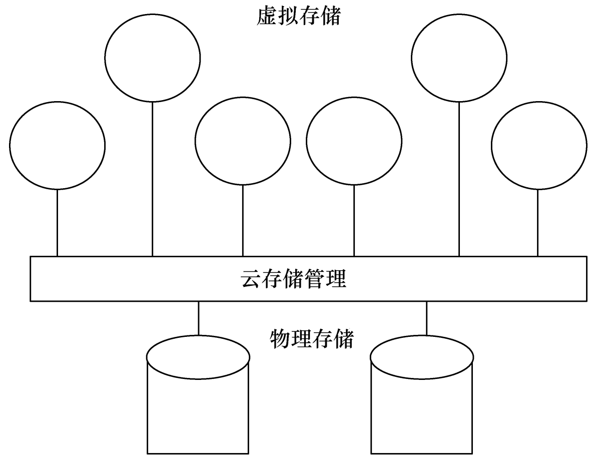


图 8.5　云存储管理

在云上运行的数据库是云数据库管理器。现有多种方法来使用云数据库管理器。在第一个模型中，为了让用户在云端运行数据库，必须购买虚拟机镜像，然后在虚拟机上运行数据库。第二个模型是数据库即服务模型，服务提供商将维护数据库。用户将利用数据库服务并支付服务费用。其中的一个例子是亚马逊关系数据库服务，它是一个 SQL 数据库服务，并具有一个 MySQL 接口（见本章参考文献［AMAZON］）。第三个模型是代表用户托管数据库的云提供商。用户可以利用云来维护数据库服务，或者可以在云上运行自己的数据库。云数据库必须优化其查询、存储和事务处理，以充分利用云提供的服务。图 8.6 为云数据管理。这里应注意到第 6 章讨论的典型服务可能用于开发云数据管理服务，这些服务可能包括云查询服务和云事务服务。

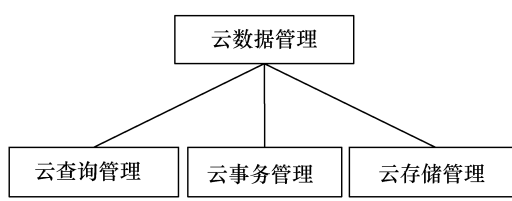


图 8.6　云数据管理

8.5　总结和展望

本章介绍了云计算的概念，并讨论了虚拟化的各个方面。特别是讨论了硬件虚拟化、操作系统虚拟化、网络虚拟化和数据库虚拟化的方面。我们还讲述了云的各种服务模型和部署模型，并简要介绍了云的功能，如存储管理和数据管理。本部分的其余章节将讲述有关云计算概念的更多详细信息。特别地，我们将说明云功能的一些细节，以及云计算系统的产品和原型。这些章节将为讨论云的安全问题以及我们开发的实验系统奠

定基础。

参 考 文 献

[AMAZON] Amazon relational database service, http://aws.amazon.com/rds/
[CLOUD] Cloud computing, http://en.wikipedia.org/wiki/Cloud_computing
[DEAN04] Dean, J. and S. Ghemawat, MapReduce: Simplified data processing on large clusters, http://research.google.com/archive/mapreduce.html
[HDFS] Apache Hadoop, http://hadoop.apache.org/
[KHAD12] Khadilkar, V., K. Y. Oktay, M. Kantarcioglu, and S. Mehrotra, Secure data processing over hybrid clouds. *IEEE Data Eng. Bull.* 35(4): 46–54, 2012.
[NIST] Definition of cloud computing, National Institute of Standards and Technology, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
[VIRTUAL] Virtualization and cloud management, http://www.vmware.com/solutions/virtualization-management/index.html

第 9 章　云计算功能

9.1　概述

在第 8 章中，我们概述了云计算中的各种概念。这包括对部署模型、服务模型以及对云中虚拟化、存储和数据管理的论述。事实上，虚拟化、存储和数据管理可以被认为是云计算功能。本章将介绍云计算功能。

我们对这些功能论述的方法是先研究计算系统的一般功能，然后再研究云对这些功能的影响。这些功能存在于操作系统、存储系统、数据库系统、信息管理系统、知识管理系统和网络系统。通过云计算功能框架将能更好地说明这些功能。

本章的组织结构如下。我们设计的云计算框架将在 9.2 节中讲述；9.3 节讨论包括虚拟化在内的操作系统功能；9.4 节讲述云网络；9.5 节说明云数据和存储管理功能；应用程序功能将在 9.6 节中论述。其他方面，如云策略管理、备份和恢复将在 9.7 节中讨论。本章总结在 9.8 节。

应该指出的是，本章仅简要讨论策略管理。当我们在第五部分中详细讨论保证云安全时，将给出策略管理的细节内容。此外，第二部分讨论的概念、标准和技术，如 Web 服务和典型数据服务可用于提供云服务（例如，云基础设施服务和云数据管理服务）。本章讨论的功能如图 9.1 所示。

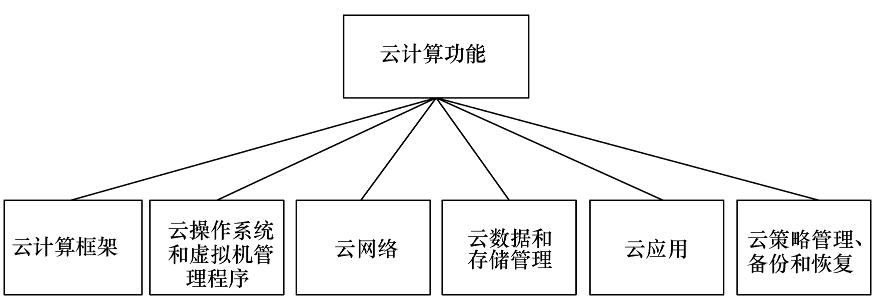


图 9.1　云计算功能

9.2　云计算框架

基于云计算功能的云计算框架如图 9.2 所示。我们定义了一个分层次的框架。最底层是网络层，最高层是应用层。应用程序可以是任何类型的应用程序，包括医疗保健、财务、防御和情报。我们在云端托管的应用可以是社交网络、内部威胁分析、恶意软件

检测、信息共享和本体管理。我们将在后面的章节中描述这些应用。还有可能有其他应用，包括信息和知识管理。

云框架的核心层是操作系统/虚拟化层、存储层和数据管理层，还包括数据挖掘功能。操作系统/虚拟化层是执行虚拟化，以及内存管理、调度和进程间通信管理的一层。存储层将管理云中大量数据的存储。我们认为 Hadoop 等分布式文件系统是管理分布式存储的存储层的一部分，对于分析任务，Hadoop 要与 Google Map/Reduce 共同解决。数据管理层将进行云查询处理、云事务管理、云元数据管理和云数据挖掘。

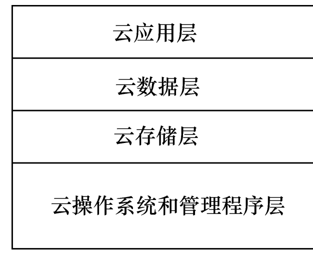


图 9. 2　云计算框架

策略管理是安全管理的一个方面，其与备份和恢复都是云计算的功能。安全性必须纳入每一层。此外，在出故障的情况下，系统必须能被恢复。我们将在后面的章节中讨论安全性。在本章的其余部分中，将讨论本节中论述的其他功能。在第 10 章我们将讨论云数据管理，在第 11 章将讲述云应用。

9. 3　云操作系统和管理程序

云基本上由一系列机器组成。每个机器（或计算机或节点）都有自己的操作系统，其称为主机操作系统。然而，为了增强云的计算能力，节点可以拥有虚拟机（Virtual Machine，VM）监视器，其也叫虚拟机管理程序。通过虚拟机管理程序，向用户提供称为客户机 OS 等许多虚拟操作系统的支持。图 9. 3 为管理程序的概念。

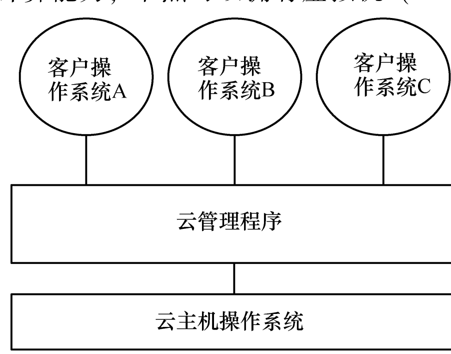


图 9. 3　管理程序的概念

克里斯本顿在关于云安全联盟的文章中，将虚拟化定义如下：虚拟化提供通过软件模拟硬件的能力（见本章参考文献［VIRT］）。该文章还指出，某些形式的操作系统仍然需要从硬件中引导。这可能是一个完整的操作系统，如 Linux，或者它可能是一个专门用于虚拟化的精简系统。在每一种情况下，首先要引导操作系统，然后加载管理程序（即 VM 监视器）。管理程序模拟客户操作系统的特定硬件配置。也就是说，当一个客户操作系统被加载到一个虚拟机中时，通过虚拟机管理程序而不是实际的硬件来模拟托管该客户机的硬件。当然，最终多个虚拟机必须映射到一台物理机器，这对于管理程序来说是一个挑战。

应该注意的是，尽管虚拟化在云中是可取的，但并不是必需的。虽然许多云使用虚拟化，特别是 IaaS 云，但基于 SaaS 的云有时不提供虚拟化功能。许多云部署时都包含虚拟化组件。虽然这是一种常见的技术，但是克里斯本顿说，人们可以将云作为多个用

户租赁的房子（即云）。房子可能有地下室（即管理程序），但是房子不是必须要有地下室。

虚拟化有两种类型的模型。一个是基于主机的，另一个是裸机。VMware 和 XEN 等系统提供基于主机的虚拟化。也就是说，管理程序在主机操作系统上运行。虽然为应用程序开发的工具可以用于管理程序，但管理程序代码量可能很大，这从安全角度来看不是很好。在裸机的情况下，操作系统支持虚拟化。这样，虽然可能没有针对管理程序而开发良好的工具，但代码量可以保持在最低限度。有关虚拟化的更多详细信息，请参阅本章参考文献［BARH03］和［VM］。

9.4 云网络

在本节中，我们将论述云网络。关于云的许多论述都侧重于虚拟化和系统方面。通常都假设商业网络足以承载云。然而，在构建云的网络方面还需要一些努力。例如，在关于云网络的论文中，作者指出，设计良好的网络来支持云架构需应对几个挑战（见本章参考文献［BORO11］)。这些挑战包括存在于 WAN（广域网）性能和安全性之间的路由优化、可靠性和延迟等问题。我们专注于前两个方面。

关于路由优化，企业要考虑的一个关键挑战是网络如何在视频通过云端传输时，来识别视频、衡量视频和区分视频优先级（见本章参考文献［BORO11］)。在私有云和混合云环境中，网络应该适应于基于用户位置、设备类型或可达性等因素来提供最佳体验。关于广域网性能方面的可靠性和延迟性，所面临的挑战是：以最大的可靠性和最小的延迟性从云端传输应用程序时，能有一致且高质量的性能。

像系统和数据虚拟化一样，云计算也正在探索网络虚拟化。如本章参考文献［BORO11］所述，网络虚拟化是将硬件、软件网络资源和网络功能组合到一个基于软件的管理实体（虚拟网络）中的过程。网络虚拟化涉及平台虚拟化，并且通常与资源虚拟化相结合。网络虚拟化被分类成外部和内部的虚拟化。在外部情况下，将许多网络或网络的一部分组合成虚拟单元；在内部情况下，向单个系统上的软件容器提供类似网络的功能。好比是用于系统虚拟化的 VM 监视器（即管理程序）的概念，虚拟网络监视器处理网络虚拟化。思科等提供商正在为网络提供虚拟化功能。在将虚拟化网络集成到云环境中还需要做更多的工作。图 9.4 为云网络。

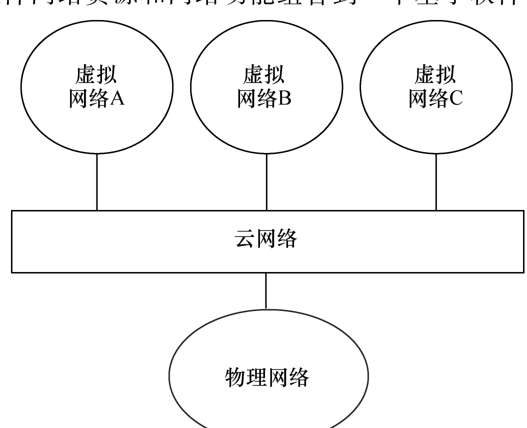


图 9.4　云网络

9.5 云数据和存储管理

最近有一些关于云数据管理的研究，我们将在第 10 章中讨论其细节。在本节中，我们将总结数据管理功能（如查询处理、事务管理及存储、元数据管理和数据挖掘）所面临的挑战。安全细节将在第五部分中给出。

关于云计算查询处理器，我们已经做了一些工作。例如，我们已经实现了使用 Hadoop、Map/Reduce 和 Hive 技术的云查询处理器，还实现了云查询处理的语义 Web 数据。相应的实现技术将在第四部分讨论。我们所面临的挑战是查询处理过程中，要提出适当的查询优化策略，这将取决于数据如何在云中跨节点存储。Kantarcioglu 已经研究了云查询处理问题。在这项工作中，他表示云查询处理的主要挑战之一是动态资源分配（见本章参考文献［OKTA12］）。

关于云中的事务处理问题，已经有学者做了一些研究。最值得注意的是约翰内斯·吉尔克（见本章参考文献［GEHR11］）所做的工作。他表示，云数据管理面临的一个重大挑战是扩展和访问。虽然基于锁的实现在比如数据库系统和搜索引擎的传统信息访问系统中占主导地位，但是出现了基于乐观并发控制机制（Optimistic Concurrency Control，OCC）的云系统。

关于云存储面临的挑战是为云环境提供适当的存储策略。在云中跨节点存储时，数据如何分片？如何利用数据虚拟化来提供最佳的存储空间？Kantarcioglu 和 Mehrotra 已经为混合云开发了引人关注的存储方案（见本章参考文献［OKTA12］）。他们指出，云计算的新兴趋势是混合云，云中终端用户站点的内部功能/资源将与云服务无缝集成，以生成强大的性价比高的数据处理解决方案。

云中的元数据管理不仅包括有关数据和云中资源的信息，还包括有关用户账户、使用模式、SLA（Service Level Agreement，服务等级协议）和其他云特定需求的信息。人们还需要做很多工作是提供适当的模型来存储和管理云中的元数据。

云数据挖掘已经成为一个重要的领域，特别是在大数据分析出现之后。数据挖掘算法相对于并行数据挖掘的性能得到了增强。现在所面临的挑战是在云上实现这些并行数据挖掘技术。需要注意的是，并行/分布式数据库与云数据库之间的差异在于，有了云可以通过迁移云数据以获得更好的资源利用率。因此，数据挖掘算法应该考虑数据的位置迁移。图 9.5 为云数据和存储管理问题。有关云中数据管理的应该关注的论文已经在本章参考文献［DMC12］中给出。

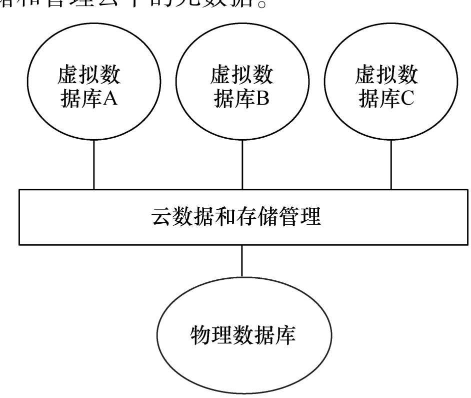


图 9.5 云数据和存储管理问题

9.6 云应用

我们需要云的各种功能去托管云上的各种应用程序。这些应用程序可能是某领域的应用程序，如医疗保健、金融或 IT 应用程序，如信息管理、知识管理以及社交网络。在本书的后续部分，我们将讨论一些已经开发的利用云的应用程序。我们的应用主要致力于云数据和信息管理上。图 9.6 说明了云应用情况，其中的一些应用将在第 11 章中讨论。

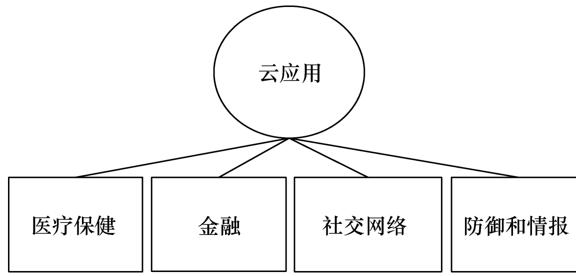


图 9.6　云应用

接下来，我们将讨论一些例子。医疗机构可能会使用云来整合众多数据源。这些数据源可以存储在云中。云可以执行比如本体结盟操作（其中涉及相同概念的多个本体被集成）和其他数据密集型操作。社交网络也可以被托管在云端。云将提供计算资源来整合和挖掘社交网络中的数据。我们将在第四部分中举例说明如何将云应用于社交网络管理和查询处理等应用。云应用程序也将在第 12 章讨论。

9.7 云策略管理、备份和恢复

我们的框架还说明了云策略管理、备份和恢复。策略管理组件将在每一层执行各种策略，这些策略可以是安全策略或其他类型的策略，如管理策略。人们已经开发了各种标准来说明和执行策略（例如，SAML、XACML、WS – Security）。

备份和恢复是云服务提供商必须执行的重要功能。云计算环境必须为客户启动和运行这些功能。因此，SLA 必须指定云要被维护多长时间。此外，为了管理上的问题，云数据必须进行备份并且在可能的情况下要进行复制。在解决云安全问题时，我们将论述一些面临的挑战，图 9.7 说明了云策略管理、备份和恢复问题。

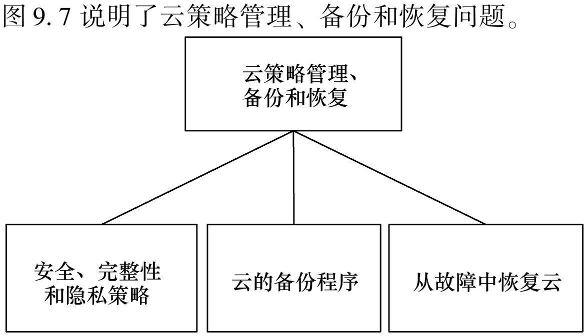


图 9.7　云策略管理、备份和恢复问题

9.8 总结和展望

本章讨论了云计算功能的各个方面。首先，我们提出了一个云计算框架。该框架由网络层、虚拟化层、存储层、数据管理层和应用层组成。其次，我们讨论了每层的功能。最后，我们还讨论了云的策略管理、备份和恢复问题。

由于云数据管理是我们的专业领域，在第 10 章我们将会讨论云数据管理。我们将在第 11 章讨论云应用。我们为云数据管理和应用开发的一些原型的细节将在本书的第四部分给出。在本书的第五部分，我们将讨论云的安全问题。特别是，我们将讨论我们的云框架中每一层的安全性。

参 考 文 献

[BARH03] Barham, P., B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, Xen and the art of virtualization. *Proceedings of the 19th ACM Symposium on Operating Systems Principles 2003* (SOSP 2003), 164–177. Bolton Landing, NY.

[BORO11] Borovick, L., and R. Mehra, White paper. *Architecting the Network for the Cloud*, January 2011 http://www.cisco.com/en/US/solutions/collateral/ns1015/white_paper_idc-architecting_the_network_for_the_cloud.pdf

[DMC12] *Proceedings of the Workshop on Data Management in the Cloud*, http://www.nec-labs.com/dm/dmc2012.

[GEHR11] Gehrke, J., *Transaction Processing in the Cloud*, White Paper, 2011.

[OKTA12] Oktay, K. Y., V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra, B., and M. Thuraisingham, Risk-aware workload distribution in hybrid clouds. *IEEE Fifth International Conference on Cloud Computing (IEEE CLOUD)*, 2012: 229–236. Honolulu, HI.

[VIRT] Cloud Security Alliance, The Basics of Virtualization Security by Chris Brenton. https://cloudsecurityalliance.org/wp-content/uploads/2011/11/virtualization-security.pdf

[VM] VMware virtualization overview. http://www.vmware.com/pdf/virtualization.pdf

第 10 章　云数据管理

10.1　概述

在过去 40 年中，数据库系统技术已经发生了很大的进步，也就是从基于网络和分层模型的派生系统，发展到基于客户端 – 服务器体系结构的关系和面向对象的数据库系统。数据库系统现在可以通过网络访问，数据管理服务也可以作为 Web 服务来实现。我们考虑一个数据库系统，同时包括数据库管理系统（DBMS）和数据库（另见本章参考文献［DATE90］中的讨论）。数据库系统的 DBMS 组件管理着数据库。数据库包含持久性数据，也就是说即使应用程序消失数据也是永久性的。

在第 9 章中，我们讨论了包括云数据管理功能在内的云功能。在本章中我们将详细介绍云数据管理。因为这个话题还处于起步阶段，我们将讨论各种数据管理功能，以及它们如何可以在云中实现。应该指出的是，“大数据”的概念越来越受欢迎。也就是说，人们必须处理大量 EB 级别甚至更多的数据。许多人认为，云计算将在大数据计划中发挥重要作用。

本章的结构如下。在 10.2 节中，我们将讨论关系数据模型及其在云上的实现。在 10.3 节中，描述了数据库系统中各种类型的架构。这些包括用于集中式数据库系统的体系结构、计划架构以及功能架构。然后，我们将讨论云上的数据管理架构。10.4 节讨论了数据库系统的功能。这些功能包括查询处理、事务管理、元数据管理和存储管理、完整性维护和容错性。需要注意的是，安全性将是本书后面几章的主题，因此本章将不再讨论安全性。对于每个功能，我们将讨论云计算环境对其的影响。数据挖掘将是 10.5 节的主题，并且会涉及云数据挖掘的内容。在 10.6 节中，我们将讨论云中的信息管理。具体来说，我们讨论云中的语义 Web 数据管理。本章在 10.7 节中进行总结。应该注意的是，第 7 章讨论的专用 Web 服务可以用于云数据管理服务，包括云存储服务、云查询服务和云事务服务。商业云数据管理产品将在第 21 章讨论。图 10.1 说明了本章中讨论的主题。

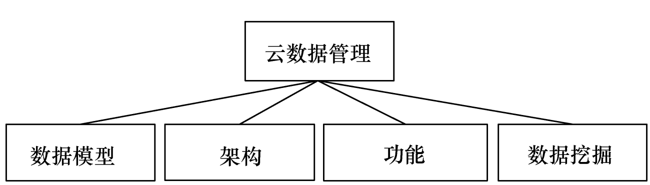


图 10.1　云数据管理

10.2　关系数据模型

一般来说，数据模型的目的尽可能准确、完整、自然地表示世界信息（见本章参考文献［TSIC82］）。本节中，我们将讨论关系数据模型的要点，因为它是当今使用最广泛的模型。使用关系模型（见本章参考文献［CODD70］），数据库被视为关系的集合。每个关系都有属性和行。例如，图 10.2 为两个关系数据库，即 EMP 和 DEPT。EMP 有 4 个属性：SS#、Ename、Salary 和 D#。DEPT 有 3 个属性：D#、Dname 和 Mgr。EMP 有三行，也称为元组，而 DEPT 有两行，每行都由其主键唯一标识。例如，SS#可以是 EMP 的主键，D#可以是 DEPT 的主键。关系模型的另一个关键特征是关系中的每个元素都是原子类型值，例如整型或字符串。也就是说，不支持像列表之类的复杂值。

EMP			
SS#	Ename	Salary	D#
1	John	20k	10
2	Paul	30k	20
3	Mary	40k	20

DEPT		
D#	Dname	Mgr
10	Math	Smith
20	Physics	Jones

图 10.2　关系数据库

各种操作都是在关系上进行的。SELECT 操作选择满足某些条件的行的子集。例如，在关系 EMP 中，可以选择工资大于 2 万（20k）的行。投影操作将关系投影到某些属性上。例如，在关系 EMP 中，可以投射到属性 Ename 和 Salary 上。连接操作对两个关系的一些常见属性进行连接。这些操作的详细讨论在本章参考文献［DATE90］和［ULLM88］中给出。

多种操作关系的语言已经被提出。在这些语言中最出名的是 ANSI（美国国家标准学会）标准 SQL（结构化查询语言）。该语言用于访问和操作关系数据库中的数据。DBMS 提供商和用户广泛接受此标准。它支持模式定义、检索、数据处理、模式操作、事务管理、完整性和安全性。其他语言有加州大学伯克利分校 INGRES 项目首次提出的关系演算（见本章参考文献［DATE90］）。关系数据库中的另一个重要概念是视图。视图本质上是虚拟关系，由数据库中的关系组成。

我们今天所面临的挑战是提出适合云端关系模型的版本。Oracle 等提供商正在构建基于云的关系数据库系统，如 MySQL 之类的系统正在云中实现。亚马逊开发了 DynamoDB 数据库。正如亚马逊所说，DynamoDB 是一个完全托管的数据库，NoSQL 数据库服务可以提供快速的、可预期的性能，并且可以实现无缝扩展。在 AWS（Amazon Web Services）管理控制台中点击几下，客户便可以启动一个新的 Amazon DynamoDB 数据库表，在不会出现停机或性能下降的情况下扩展或降低对表的请求容量，并且也可以查看资源利用率和性能指标。研究人员正在研究在云中托管这种数据库的方法。另外，专门

为云开发的专用数据模型也正在探索中。

10.3 体系结构问题

本节介绍数据库系统的体系结构，并研究云的影响。首先，我们说明一个数据库系统的集中式体系结构。然后，我们描述一个分布式数据库体系结构。图 10.3 是集中式体系结构的一个例子。这里，DBMS 是一个单一的实体，它管理一个集中的数据库。功能架构说明了 DBMS 的功能模块。DBMS 的主要模块包括查询处理器、事务管理器、元数据管理器、存储管理器、完整性管理器和安全管理器。集中式数据库系统体系结构（见图 10.3）的 DBMS 组件的功能架构如图 10.4 所示。

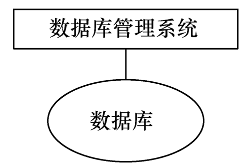


图 10.3　集中式体系结构

虽然已经给出了分布式数据库系统的许多定义，但是还没有标准定义。我们对分布式数据库系统概念和问题的讨论受到本章参考文献［CERI84］的影响。分布式数据库系统包括 DDBMS、分布式数据库和用于通信的网络。DDBMS 管理分布式数据库。分布式数据库是分布在多个数据库中的数据。我们选择的分布式数据库系统体系结构是一个多数据库架构，它是紧密耦合的。该体系结构如图 10.5 所示。选择这样的架构，就可以解释基于这种方法的同构和异构系统的概念。在此架构中，节点通过通信子系统连接，本地应用程序由本地 DBMS 处理。此外，每个节点还涉及至少一个全局应用程序，因此在此架构中没有集中控制。DBMS 通过称为分布式处理器（DP）的组件连接。在同构环境中，本地 DBMS 是同构的，而在异构环境中，本地 DBMS 可能是异构的。

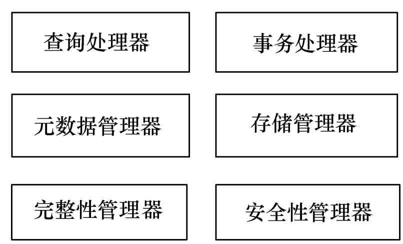


图 10.4　DBMS 的功能架构

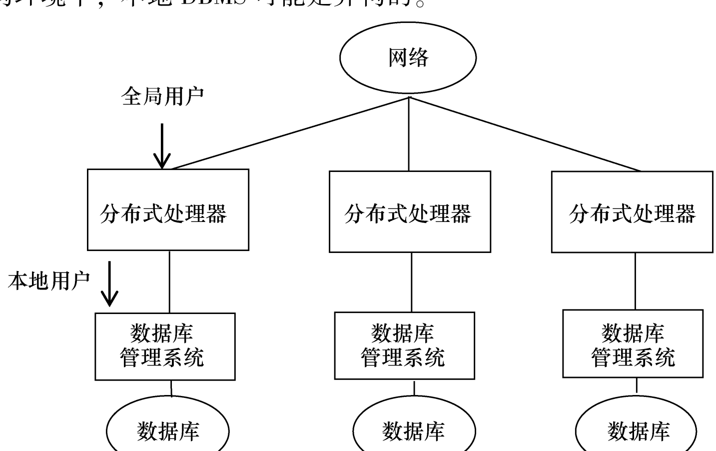


图 10.5　一个 DDBMS 体系结构

分布式数据库系统功能包括分布式查询处理、分布式事务管理、分布式元数据管理以及跨多个节点实施安全性和完整性。分布式处理器（Distributed Processor，DP）是DDBMS的重要组成部分，这个模块连接本地不同的DBMS。也就是说，每个本地DBMS均由DP扩充。DP的模块如图10.6所示。其组件有分布式元数据管理器（Distributed Metadata Manager，DMM）、分布式查询处理器（Distributed Query Processing，DQP）、分布式事务管理器（Distributed Transaction Manager，DTM）、分布式安全管理器（Distributed Security Manager，DSM）和分布式完整性管理器（Distributed Integrity Manager，DIM）。DMM管理全局元数据。全局元数据包括有关模式的信息，其中描述了分布式数据库中的关系、关系分段的方式、分段的位置以及强制执行的约束；DQP负责分布式查询处理；DTM负责分布式事务管理；DSM负责执行全局安全约束；DIM负责在全局层面保持完整性。应注意的是，DP的模块与远程节点的对等体通信。例如，节点1处的DQP与节点2处的DQP进行通信，用于处理分布式查询。

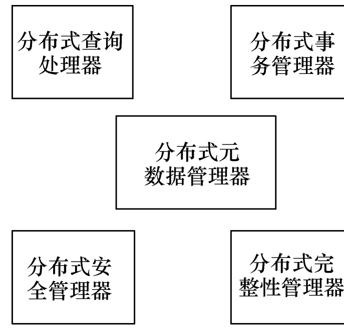


图10.6　DP的模块

研究人员面临的挑战是为云提供一个合适的体系结构。例如，在集中式体系结构的情况下，所有数据都可以托管在一个服务器上，也可以在多个服务器上进行处理。这种方式不能充分利用云提供的功能。在分布式数据库情况下，数据库的每个片段都可以存储在服务器上。虽然这可能会更好地利用云，但它并没有提高资源利用率。云数据库体系结构必须有效地使用资源利用率，并以最大限度存储数据。包括IBM、Oracle和Microsoft在内的多个数据库供应商正在探索云数据管理架构。

10.4　DBMS功能

10.4.1　简介

DBMS的功能架构如图10.4所示（另见本章参考文献［ULLM88］），DBMS的功能是执行其操作。DBMS从根本上管理数据库，并能够支持用户查询和更新数据库。因此，DBMS的基本功能是查询处理和更新处理。在某些应用程序中，例如银行，查询和更新是作为事务的一部分发布的。因此，事务管理也是DBMS的另一个功能。为了执行这些功能，必须保持关于数据库中的数据的信息，此信息称为元数据。与管理元数据相关的功能是元数据管理。需要专门的技术来管理实际存储数据的空间，与管理这些技术相关的功能是存储管理。为了确保上述功能正确执行，保证用户获取数据的准确性，我们还需要一些另外的功能，这些包括安全管理、完整性管理和故障管理（即容错）。

本节重点介绍DBMS的一些核心功能，分别是查询处理、事务管理、元数据管理、存储管理、完整性维护和容错。这些功能将在10.4.2～10.4.7节讨论。在10.4.6节

中，还将讨论云的影响。图 10.7 说明了本节中讨论的概念。

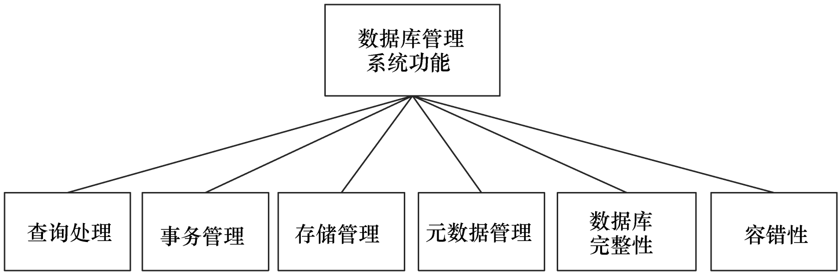


图 10.7　数据库管理系统功能

10.4.2　查询处理

查询操作是 DBMS 中最常用的功能。用户可以查询数据库并获得查询结果。查询处理有以下两个方面：一方面是需要一个好的查询语言。SQL 等语言对于关系数据库来说很受欢迎。这种语言正在扩展用于其他类型的数据库。另一方面是用于查询处理的技术。人们已经提出了许多算法用于查询处理，特别是对于 JOIN 操作。另外，不同的策略也可以执行一个特定的查询。首先计算各种策略的开销，然后通常选择开销最低的策略用来执行，这个过程称为查询优化。开销通常由磁盘访问决定，因此我们的目标是在处理查询时应最大限度地减少磁盘访问。

用户往往使用他们喜欢的一种语言进行查询。该语言的结构必须被转换成数据库系统所理解的结构，这个过程称为查询转换。可以基于不同模式分阶段进行查询转换。例如，基于外部模式的查询，首先被转换为关于概念模式的查询，然后将其转换为物理模式的查询。一般来说，转换过程中使用的规则包括通用子表达式的分解，并尽可能地在查询树中将选择和投影向下推送。如果在连接之前执行选择和投影，连接的开销就可以减少很多。

查询处理中的模块如图 10.8 所示。用户接口管理器接受查询，解析查询，然后将其提供给查询转换器。查询转换器和查询优化器相互通信以产生执行策略。存储管理器用来访问数据库。响应管理器向用户做出响应。

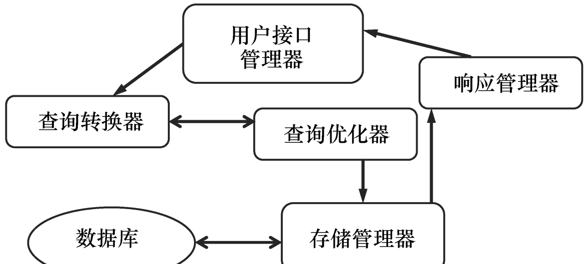


图 10.8　查询处理器

有关云查询处理的工作已经开展。例如，我们已经实现了使用 Hadoop、Map/Reduce 和 Hive 技术的云查询处理器，具体实现技术将在本书的第四部分讨论。提出对查询处理相应的优化策略，是实现的关键。这将取决于数据如何在云中的节点之间进行存储。有的研究人员还研究了针对云中面向列的数据库，并宣称这提供了更高的性能。在这样的数据库中，关系根据列分段，即关系的每列构成对应的表。Composite Software 公司的研究人员利用数据虚拟化技术为云开发了高效的查询算法。他们指出，为了实现高效和最佳查询执行，数据虚拟化使用的策略包括基于成本的查询计划评估和选择、基于规则的查询计划修改、数据源能力分析、连接算法种类和配置、选择池和传播、不必要操作的精简、流水线结果流、并行扫描执行和检索、限制操作员预处理和冗余扫描复用等。这些基本上都要利用云，并且必须充分利用并行查询处理。

Kantarcioglu 等人已经研究了云查询处理的问题。在他们的工作中，云查询处理的主要挑战之一是动态资源分配（见本章参考文献［OKTA12］）。与传统环境不同，云基础设施通常允许应用程序动态增加或减少计算资源的数量。这可能对查询处理和优化的处理方式有重要的影响。例如，如果我们有服务约束的质量保证来响应查询，我们可以动态添加新的计算资源来处理更多的数据。同时，一旦我们不需要新的计算资源，我们就可以减少计算资源的使用。显然，这种动态分配需要一些开销。例如，在 Amazon EC2 上，使用更多的机器意味着会为这些机器支付更多的费用。此外，加载这些机器的镜像，并把必要的数据移动到新启动的机器上将需要一些时间。因此，当查询优化完成时，就应考虑此类开销。为了解决这些问题，作者已经修改了现有的查询优化技术，以在优质服务约束中包含了动态资源分配开销。

10.4.3　事务管理

事务是一个程序单元，必须完全执行或根本不执行。如果事务串行执行，那么就会出现性能瓶颈。因此，事务要并发执行。当多个事务更新数据库时，适当的技术必须确保数据库是一致的。也就是说，事务必须满足 ACID（原子性、一致性、隔离和耐久性）属性。事务管理的主要方面是可串行性、并发控制和恢复。我们在本节中对此进行简要讨论。关于事务管理的详细讨论，可以参考本章参考文献［KORT86］和［BERN87］。

可串行性：调度是由多个事务执行的一系列操作。如果两个调度的结果相同，则两个调度等价。串行调度是不同时执行两个事务的调度。事务管理的目标是确保任何调度等价于一个串行调度。这样的调度被称为可串行化的调度。现在已经为 DBMS 制定了用于测试调度的可串行性的各种条件。

并发控制：并发控制技术确保当多个事务同时更新数据库时，数据库要保持一致。确保调度可串行化的三种流行并发控制技术是加锁、时间戳和验证（也称为乐观并发控制）。

恢复：如果事务由于某些故障而中止，则必须使数据库与初始状态处于一致，这是事务恢复。处理事务故障的一个解决方案是维护日志文件。事务的动作记录在日志文件

中。因此，如果事务中止，则通过撤销事务的操作就将数据库恢复到一致状态。撤销操作的信息在日志文件中。另一个解决方案是记录事务的动作，但不会对数据库进行任何更改。只有当事务提交时，数据库才能被更新。这意味着日志文件必须被稳定地存储。另外，对上述技术的各种修改也已经被提出来，用于处理不同的情况。

当事务在多个数据源中执行时，则使用称为“两阶段提交”的协议来确保多个数据源是一致的。图 10.9 说明了事务管理的各个方面。

关于云中事务处理的一些工作已经开展，最引人注目的是约翰内斯·格尔克（Johannes Gehrke）。他表示，云数据管理所面临的主要挑战是扩展和访问（见本章参考文献［GEHR11］）。虽然基于锁的实现在传统信息访问系统（比如数据库系统和搜索引擎）中占主导地位，但是出现了基于乐观并发控制（Optimistic Concurrency Control，OCC）的云系统。然而，我们不知道基于锁或基于 OCC 的信息访问在云中是否有优势，没有系统的科学研究可以给云系统设计人员一个准确的答案。格尔克和他的同事正在研究基于锁和 OCC 的新型云架构。随着事务率的增加，这种架构平稳扩展（资源占用与系统负载成比例），并可与云查询管理器无缝连接，而且可通过查询和事务实现不同级别的隔离。

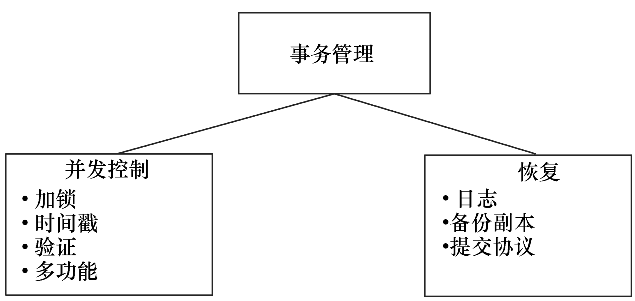


图 10.9　事务管理的一些方面

10.4.4　存储管理

存储管理器负责访问数据库。为了提高查询和更新算法的效率，必须执行适当的访问方法和索引策略。也就是说，在生成执行查询和更新请求的策略时，需要考虑使用的访问方法和索引策略。用于访问数据库的访问方法将取决于索引方法。因此，创建和维护适当的索引文件是数据库管理系统中的一个重要问题。通过使用适当的索引机制，查询处理算法可不必搜索整个数据库。相反，可以直接访问将要检索的数据。因此，检索算法更有效率。图 10.10 为一个索引策略的示例，其中数据库根据项目进行索引。

人们对关系数据库系统开发适当的访问方法和索引策略进行了大量研究。10 – 树和哈希（见本章参考文献［DATE90］）是索引策略的示例。目前的研究重点是开发面向对象数据库系统的这种索引机制，以支持多媒体数据以及 Web 数据库系统等。

为云提供适当的存储策略，是开发人员和研究人员面临的挑战。数据如何分散在云

中的不同节点上？如何利用数据虚拟化来提供最佳的存储空间？Kantarcioglu 和 Mehrotra 为混合云开发了引人关注的存储方案。在本章参考文献［OKTA12］中他们指出，云计算的新兴趋势是混合云，云中终端用户站点的内部功能/资源将与云服务无缝集成，以生成强大的、经济高效的数据处理解决方案。混合云的解决方案为云计算提供了类似的优势，例如，高输出时使用公共资源的能力对于即付即用（而不是为内部峰值负载创建基础设施的资本成本）的模式非常重要。然而，它们提供了更好地控制成本的机会（例如，如果本地资源足够，所使用的云资源就最小化），并且更好地管理数据的隐私和机密性（例如，对什么信息和处理可以公开于公有云上进行控制）。例如，最终用户可以根据私有云上的敏感信息运行关键任务，同时将日常任务外包给公有云。Kantarcioglu 及其同事正在探索解决混合云背景下的信息管理挑战，并提出解决这些挑战的新颖解决方案。

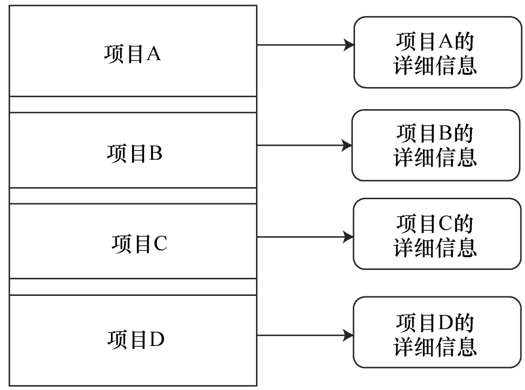


图 10.10 项目索引实例

Kantarcioglu 和 Mehrotra 指出，混合云中的信息管理可能以多种方式利用公共资源。例如，可以在公共和私有系统之间划分查询工作量，以通过利用查询间的并行性来最大化系统的吞吐量。在给定查询（内部查询并行性）的环境中，也可以进一步利用混合云提供的固有并行性，即使这样的策略可能导致额外的网络成本。在考虑混合云架构中的各种策略时，我们还需要考虑成本和资源限制。云服务提供商通常支持有竞争力的成本模型，并为终端用户提供不同的服务等级协议（SLA）来进行数据存储和处理。例如，AWS 提供了一种分层定价模式，使用数据和处理服务越多，价格就越便宜。AWS 还为弹性计算云（Elastic Compute Cloud，EC2）或简单存储服务（Simple StorageService，S3）提供 SLA，如果亚马逊不能达到至少 99% 的时间承诺，那么就要返回给用户 10% ~25% 的月费。类似的分层定价模型和 SLA 也由其他云服务提供商（例如，Microsoft Windows Azure 和 Google App Engine）提供。利用这种混合云架构的信息管理系统存在几个关键问题。第一个问题是如何将信息存储在混合云中？一个好的解决方案必须考虑到所选择的数据表示策略、与存储基础设施相关的经济成本和查询工作量特征。第二个问题是在这种分布式架构中的有效查询处理。他们正在通过为混合云开发适当的存储模型来探索解决这些问题。

10.4.5 元数据管理

元数据描述数据库中的数据。例如，在图 10.2 所示的关系数据库的情况下，元数据包括以下信息：数据库有两个关系分别为 EMP 和 DEPT。EMP 有 4 个属性，DEPT 有

3 个属性等。主要问题之一是为元数据开发一个数据模型。在我们的示例中，也可以将关系模型套用于元数据中。图 10. 11 所示的元数据关系 REL 由关系和属性的信息组成。

关系REL	
关系	属性
EMP	SS#
EMP	Ename
EMP	Salary
EMP	D#
DEPT	D#
DEPT	Dname
DEPT	Mgr

图 10. 11　元数据关系

除了关于数据库中数据的信息之外，元数据还包括有关访问方法、索引策略、安全约束和完整性约束的信息。还可以将策略和程序作为元数据的一部分。一旦定义了元数据，就会出现管理元数据的问题。查询和更新元数据的技术是什么？由于所有其他 DBMS 组件为了进行处理问题都需要访问元数据，那么元数据管理器和其他组件之间的接口是什么？对于关系数据库系统，元数据管理比较好理解。当前的挑战在于如何管理更复杂系统的元数据，比如数字图书馆和网络数据库系统等。

云中的元数据管理不仅包括有关数据和云中资源的信息，还包括有关用户账户、使用模式、SLA 和其他云特定的信息。因此，使用适当的模型来存储和管理云中的元数据还需要做大量的工作。

10. 4. 6　数据库完整性

并发控制和恢复技术可以保持数据库的完整性。此外，还有另一种类型是强制完整性约束来保持数据库的完整性。在数据库系统中强制执行的完整性约束有两种类型：独立于应用的完整性约束和特定于应用程序的完整性约束。完整性机制还包括确定数据质量的技术。例如，数据和数据源的准确度是多少？保持数据质量的机制是什么？输出数据的准确度如何？关于基于数据质量的完整性的讨论，请参考本章参考文献［DQ］。需要注意的是，数据质量对数据挖掘和数据仓库来说非常重要。如果挖掘的数据不好，则不能信赖其结果。独立于应用程序的完整性约束包括主键约束、实体完整性规则、参照完整性约束以及归一化过程中涉及的各种功能依赖性（参见本章参考文献［DATE90］中的讨论）。应用程序特定的完整性约束是特定于应用程序的约束。示例包括“员工的工资不能减少”和“没有经理可以管理两个以上的部门”。人们已经提出了各种技术来强制执行特定于应用程序的完整性约束。例如，当更新数据库时，将检查这些约束并检验数据。数据库完整性包括的方面如图 10. 12 所示。

目前已经开展了相关云中数据库完整性的一些研究工作，这些工作基于数据来源和类别。例如，云中的数据可能会在一段时间内来自多个来源。因此，保持数据的来源至关重要。其他挑战包括执行适当的完整性约束。也就是说，如何在数据虚拟化的云端执行完整性约束？人们仍需要对云中的数据完整性进行大量研究。

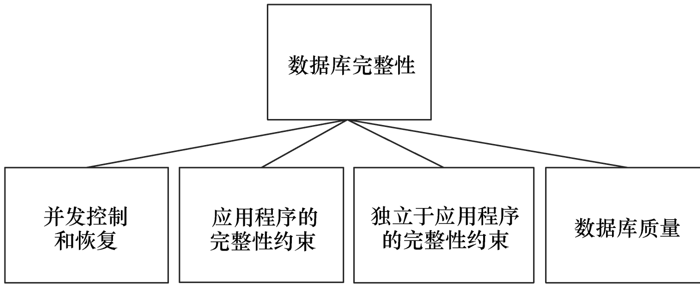


图 10.12　数据库完整性的一些方面

10.4.7　容错

前面两节讨论了数据库的完整性和安全性，与其密切相关的特征是容错。我们几乎不可能保证数据库将一直按计划运行。实际上可能会发生各种故障。这些可能是硬件故障或软件故障。如前所述，事务管理的主要问题之一，是确保数据库在存在故障的情况下恢复到一致状态。所提出的解决方案包括维护适当的日志文件记录事务的动作，以预防出现其动作必须被撤销的情况。

处理故障的另一种方法是检查点技术。在数据库处理过程中会建立各种检查点。在每个检查点，确保数据库处于一致状态。因此，如果在处理过程中出现故障，则必须将数据库恢复至上一个检查点的正常状态。这样可以保证数据库是一致的。与检查点密切相关的是验收测试。在各种处理步骤之后，进行验收测试。如果通过测试，那么它们就可以进行下一步。容错包括的内容如图 10.13 所示。

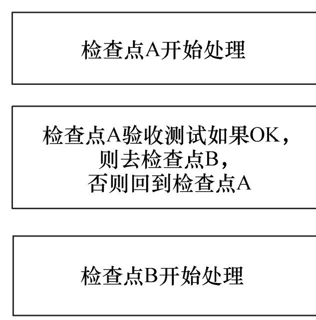


图 10.13　容错包括的内容

确保云端持续运行是云服务提供商面临的主要挑战之一。SLA 通常指定云的运行方式以及维护时间。因此，确保云数据库的备份和开发适当的容错解决方案至关重要。这个话题在我们处理值得信赖的云时，也就是在本书的第八部分进一步讨论。

10.5　数据挖掘

数据挖掘是从各种查询和数据库存储的大量数据中提取有用的信息、模式和以前未知的趋势的过程。实际上，对于许多组织机构来说，数据挖掘的目标包括改进营销能力、检测异常模式，并根据以往经验和当前趋势预测未来。很显然这种技术是需要的。随着大量当前和历史数据被存储，数据库变得越来越大，因此越来越难以支持决策制定。另外，数据可能来自多个信息源和多个域。因此，分析数据来支持企业的规划和其

他功能也是需要的。这种分析数据的技术就是数据挖掘。目前各种术语被用来指代数据挖掘，这些术语包括知识/数据/信息发现和知识/数据/信息提取。需要注意的是，有些人定义数据挖掘为提取以前未知信息的过程，而知识发现被定义为解读已经提取的信息的过程。

一些数据挖掘技术包括基于统计推理技术、归纳逻辑编程、机器学习、模糊集和神经网络的技术。数据挖掘的结果包括分类（查找将数据分成组的规则）、关联（查找数据之间的关联规则）和排序（查找排序的规则）。从本质上说，人们获得了一些假设，即从所观察到的例子和模式中提取的信息。这些模式从一系列查询中可被观察到，每个查询依赖于从先前提出的查询获得的响应。数据挖掘已经有了几个发展阶段。各种工具的论述在本章参考文献［KDN］中给出。在本章参考文献［AGRA93］和［BERR97］中很好地给出了对结果和技术的讨论。

数据挖掘是多种技术的集成。这些包括数据管理，如数据库管理、数据仓库、统计、机器学习、决策支持以及比如可视化和并行计算之类的数据管理。数据挖掘有一系列步骤。这些包括获取要挖掘的数据、确定挖掘的结果、选择挖掘的工具、进行挖掘、精简结果，以使有用的数据被进一步考虑，从挖掘中采取措施，并评估这些行为带来的好处。

对于数据挖掘，虽然已经取得了一些进展，但仍然面临许多挑战。例如，由于数据量大，算法将如何确定要选择哪种技术，以及要采取哪种类型的数据挖掘？此外，数据可能不完整或不准确。有时可能有冗余的信息，有时也可能没有足够的信息。还需要有可以切换多种技术、并支持多种结果的数据挖掘工具。关于数据挖掘的一些当前的研究趋势，包括挖掘网络数据、挖掘分布式和异构数据库，以及隐私保护数据挖掘，它可以确保人们从挖掘中获得有用的结果，同时又保护个人的隐私。图 10.14 为数据挖掘包括的各个方面。

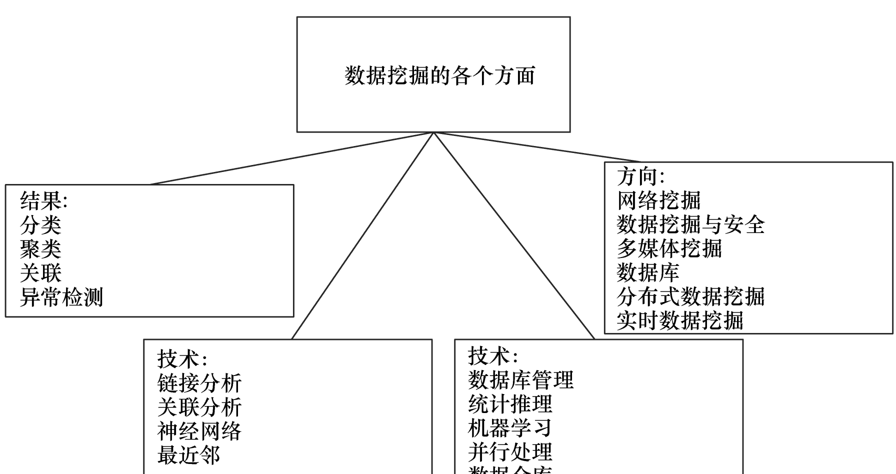


图 10.14　关于数据挖掘的各个方面

随着大数据分析的出现，云数据挖掘正在成为一个重要的领域。相对于并行数据挖掘的性能，数据挖掘算法研究得到了增强。这里的挑战是在云上实现这些并行数据挖掘技术。需要注意的是，并行/分布式数据库与云数据库之间的差异在于：通过云，可以迁移数据以获得更好的资源利用率。因此，数据挖掘算法应该考虑数据的位置迁移。

10.6 其他方面

网络用户的爆炸式增长，以及万维网服务器数量的日益增加，正在迅速推动网络上数据库管理的发展。因此，各种各样的信息源必须被集成，以便用户以透明和及时的方式访问服务器。增加人类推理、解释和决策能力的需求导致了语义 Web 的出现，这是一个尝试将网络从目前的、只有人类可读的形式转变为机器可处理形式的措施。已出现的语义 Web 技术被用来管理网络和云端的大量数据。这些技术包括 XML、RDF 和 OWL。

我们开发了用于管理云中语义 Web 数据的实验系统，还开发了语义 Web 数据的查询优化器。这些查询优化器已在云上实现。此外，我们正在研究在云中进行管理的社交网络。本书将在第四部分中讲述在云中查询处理与语义 Web 数据相关的实验工作。语义云是利用语义 Web 技术（例如，管理大量的语义 Web 数据以及执行以语义 Web 语言表示的策略）的云。可以利用第 6 章中提到的语义 Web 服务概念为消费者提供语义云服务。

云也适合提供知识管理服务。越来越多的知识来自不同数据源的在线分析，例如组织机构数据、大规模监测或调查数据以及社交数据流。通常，分析的数据集非常大。另外，它们通常由组织机构内的不同组收集，由不同的管理人员和访问策略管理，并且维护在不同类型的数据存储库（例如，关系数据库、RDF 存储）中。此外，它们分布在多个站点之间，每个站点都由其自己的计算集群支持。因为数据量的大小和与之相关的管理策略，采取在一个位置存储所有数据的方法是不切实际的。此外，公司的知识管理应用程序（如专家查找器）可能需要存储大量数据和进行大规模计算。因此，这样的操作可能被外包给云。

云也对我们在第 7 章中讨论的其他方面（如活动管理）产生影响。活动管理（如协作和工作流以及电子商务）都可以在云中进行。然而，由于电子商务应用程序可能包含信用卡信息等敏感数据，所以云中的安全性对于成功执行电子商务和其他需要处理敏感数据的应用程序至关重要。对于这样的应用，混合云似乎是一种合适的方法，其中敏感数据存储在私有云中，而非敏感数据存储在公有云中。

我们的兴趣在于解决跨多个不同处理集群的数据分布式查询，特别是能够支持大数据分析。目前正在专门研究在封闭环境中的集群组，例如研究在一个单一的组织内而不是网络上的一般分布式查询处理。当前的任何技术都不支持包含非常大的数据存储的跨集群查询处理。目前，没有通用的方法能以可以促进分布式查询处理器对集群中的数据进行汇总或编目。由于集群之间网络连接得相对较慢，这就要求使用数据通信高效的策略，来高效地访问和组合各个站点中的数据。最后，虽然存在用于 SQL 上下文中的重组分布式查询的有效机制，但是分布式集群不一定是关系数据库，因此有必要开发不同

的策略来收集从非关系数据源检索到的数据片段。这里的研究挑战包括查询分解和优化以及响应配置。

10.7　总结和展望

本章讨论了数据管理系统的各个方面，并研究了云对它们的影响。特别是讨论云查询处理、云事务管理和云存储管理等方面。我们还讨论了云中的信息管理、知识管理和活动管理。第 7 章中讲述的针对数据、信息、知识和活动管理的专用服务的概念，可以适用于云数据信息、知识和活动管理。此外，我们还讨论了如何通过语义云提供语义 Web 服务。

如前所述，医疗、金融和国防等领域的许多应用需要处理敏感数据。因此，将安全性纳入云计算将是至关重要的。在本书第五部分和第六部分中，我们将讨论云的安全性以及我们开发的云安全实验系统。我们还将讨论云数据管理功能的安全性。在第 11 章中，我们将讨论包括社交网络云服务，以及知识管理和领域应用在内的云应用。此外，还将讨论一些专用云，如多媒体云和移动云。云产品将在第 12 章讨论。

参 考 文 献

[AGRA93] Agrawal, A. et al., Database mining: a performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6), 914–925, 1993.

[BERN87] Bernstein, P. et al., *Concurrency Control and Recovery in Database Systems*, Addison-Wesley, MA, 1987.

[BERR97] Berry, M. and G. Linoff, *Data Mining Techniques for Marketing, Sales, and Customer Support*, John Wiley, NY, 1997.

[CERI84] Ceri, S. and G. Pelagatti, *Distributed Databases, Principles and Systems*, McGraw-Hill, NY, 1984.

[CODD70] Codd, E. F., A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 1970.

[COMP] Composite software for cloud data integration, http://www.compositesw.com/solutions/cloud-data-integration/

[DATE90] Date, C., *An Introduction to Database Systems*, Addison-Wesley, Reading, MA, 1990.

[DQ] *MIT Total Data Quality Management Program*, http://web.mit.edu/tdqm/

[GEHR11] Gehrke, J., Transaction processing in the cloud, White Paper, 2011.

[KDN] Kdnuggets, http://www.kdn.com

[KORT86] Korth, H. and A. Silberschatz, *Database System Concepts*, McGraw-Hill, NY, 1986.

[OKTA12] Oktay, K. Y., V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra, B. M. Thuraisingham, Risk-aware workload distribution in hybrid clouds. *Proceedings of IEEE Fifth International Conference on Cloud Computing (IEEE CLOUD)*, Honolulu, HI, 2012: 229–236.

[TSIC82] Tsichritzis, D. and F. Lochovsky, *Data Models*, Prentice-Hall, NJ, 198B.

[ULLM88] Ullman, J. D., *Principles of Database and Knowledge Base Management Systems,* Volumes I and II, Computer Science Press, MD, 1988.

第 11 章　专用云、服务和应用

11.1　概述

我们前面讲过云计算的概念、功能和产品，还详细讨论了云数据管理功能、云数据管理产品以及云框架。另外，我们还对云服务提供商进行了概述。对云计算概念的讨论包括云部署模型以及云服务模型。也就是说，我们基本上讲述了常用的云计算概念。另外，还讲述了一些常用功能，如查询处理网络和应用程序管理。

在本章中，我们将探讨云的一些专用服务和应用。例如，近年来，手机、笔记本电脑等移动设备的使用，以及最近的智能手机都有爆炸式发展。这些设备最初专注在一些简单的应用程序，如发送和接收电子邮件，以及拨打电话。现今，这些设备专注于高度复杂的应用，例如照片共享、视频消息传递和执行过程密集计算。这些智能手机应用程序迫切需要访问云进行存储和处理数据，数据可以是流视频数据。因此，一些专用云正在出现，这些包括移动云和多媒体云。此外，云还在许多领域被用于各种应用。

在本章中，我们将概述一些专用云。在 11.2 节中，我们将讨论移动云和多媒体云。在 11.3 节中，我们将讨论云应用，如医疗保健、国防和情报、金融和社交网络。11.4 节总结本章内容。应该注意的是，我们在第 7 章中讨论的一些专用服务可以用于提供云应用服务，这些包括知识管理、社交网络和域名应用服务。图 11.1 为本章讨论的概念。

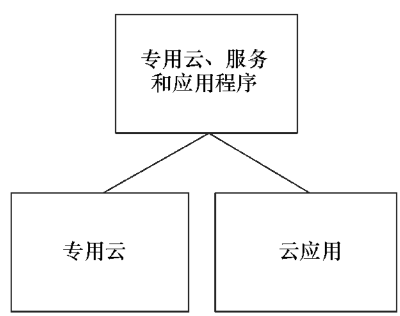


图 11.1　本章讨论的概念

11.2　专用云

在本节中，我们将讨论专用云，如图 11.2 所示。

11.2.1　移动云

如本章参考文献［MCC］所述，移动云计算是云计算与移动设备相结合使用。移动设备（如 iPhone 和 Android）具有网页浏览功能。因此，移动云计算为移动网络提供商（如沃达丰、Orange 和 Verizon）创造了机会。例如，Verizon 已经进行了大量收购（例如，Terremark 和 CloudSwitch）来增强其移动计算产品。

移动云计算还使智能手机的用户将其数据存储在云端。例如，关于移动计算的开创性发展之一是 Android 操作系统。如本章参考文献［ANDROID］所述，Android 是基于 Linux 的操作系统，适用于智能手机和平板电脑等移动设备。由 Google 和开放手持设备联盟开发与领导。其他值得注意的智能设备包括苹果公司的 iPhone 和 iPad。亚马逊网络服务（Amazon Web Services，AWS）使得开发人员可以用云将移动应用程序直接集成到苹果公司的 iPhone、iPad 和 iPod Touch 以及基于 Android 的智能手机中。如本章参考文献［AWS］所述，亚马逊的目标是使开发人员更轻松地构建利用其基于云服务的移动应用程序。基本上，有了这样的能力，开发人员可以将其应用程序与 Amazon 的基于云的简单存储服务（Simple Storage Service，S3）和 SimpleDB 数据库进行集成，并使用简单通知服务（Simple Notification Service，SNS）和简单队列服务（Simple Queue Service，SQS）发送消息。

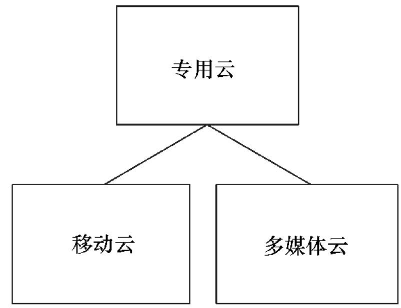


图 11.2　专用云

其他云操作系统包括阿里巴巴（中国公司，且已经发布了叫阿里云的智能手机操作系统）公司开发的操作系统。阿里巴巴公司表示，阿里云（Aliyun）是智能手机的第一个云操作系统。然而，在 IBM 开发人员（见本章参考文献［IBM］）的文章中，据称 Android 操作系统是基于 Linux 架构的云计算的理想选择。他们指出，便携式设备补充强大服务器的云计算，需要一个操作系统，这个操作系统应该可以最大限度地增加服务器架构师和程序员在小型客户端机器上可以做的事。毫无疑问，Android 就是这样的操作系统。

智能手机、移动操作系统，尤其是移动云正处于发展阶段。这个行业需要一段时间才能成熟，而且我们将看到这个市场上赢家的出现。尽管如此，随着数十亿智能手机用户的到来，移动云预计将成为云计算的主要发展方向之一（需要注意的是，2011 年第四季度智能手机全球销售额接近 1.6 亿）。

11.2.2　多媒体云

新兴的另一种专用云是多媒体云。微软亚洲研究学院的朱文武博士介绍了几种类型的多媒体云，包括多媒体感知云（Media Cloud，云媒体）和云感知多媒体（Cloud Media，媒体云）（见本章参考文献［ZHU11］）。他指出，媒体云解决了云如何执行分布式多媒体处理和存储的问题，以及为多媒体服务提供服务质量（Quality of Service，QoS），而云媒体解决了多媒体服务和应用程序如何最佳地利用云计算资源来实现更好的体验质量（Quality of Experience，QoE）。他认为应用实例是电子医疗监控、全景相机和自由视角视频。

丹麦的研究人员开发了一个名为 i5cloud 的系统，这是一个移动多媒体云。他们表示，移动多媒体云（i5Cloud）的目标是为移动多媒体和大规模社交网络分析领域的各种服务和应用提供 IaaS 和 PaaS。他们的 IaaS 提供虚拟化，并利用 Sun Solaris Container。

PaaS 包含流媒体管理器。在应用服务层和平台之间，具有多媒体业务层，其包括协同多媒体业务和视频业务。

实际上，多媒体云不仅为海量的多媒体数据（包括视频、音频、动画、地图、照片和文本）提供存储功能，还可以向数据用户提供多媒体数据的实时流式传输。用户可以是桌面用户或移动设备的用户。因此，面临的挑战是将云计算服务与移动计算服务和多媒体计算服务相结合。就像十年前一样，目标是将网络与移动网络和多媒体网络进行整合，而今天的挑战是开发移动多媒体云。

11.3 云应用

云可以用于任何应用，包括国防和情报、医疗保健和金融、会计等，它们也可以用于 IT 应用，包括网络安全、知识管理、社交计算以及其他数据和进程密集型应用。我们将讨论每个应用程序。图 11.3 说明了这些应用。

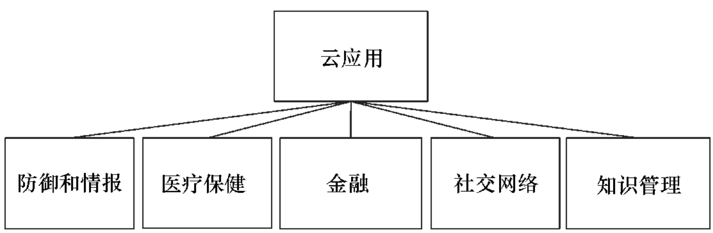


图 11.3 云应用

防御和情报：收集大量关于各种作战行动，以及监视和侦察行动的数据。这些数据必须快速处理，以便及时做出适当的决定。云计算不仅可以用于存储大量数据，还可以处理数据（如挖掘数据），以便将结果发送给决策者。这里主要的挑战是操作的安全性。大部分数据可能是高度机密的。而且，一旦处理不能被篡改。因此，这些数据必须由私有云管理和处理。

医疗保健：收集大量数据，包括患者数据和管理数据，用于存储和处理。可以分析患者数据以确定患者的状况，使医生看过之后可以确定给患者使用适当的药物。隐私对患者数据至关重要。此外，操作的安全性也至关重要。这里还需要一个私有云来存储和处理数据。

金融：银行可以对他们的数据财务处理进行外包。例如，客户账户管理和直接处理应用程序可能会外包给云。然而，客户数据必须加密，这意味着它不能存储在公有云上。此外，重要的银行业务不能外包给公有云，因为任何错误都可能对银行造成灾难性的影响。

社交网络：在这里，像 Facebook 这样的组织可以在云端托管大量的社交网络。由于个人输入的信息通常不被视为私人信息，所以这种社交网络可以利用公有云。但是，仍有隐私问题。这是因为 Facebook 有朋友的概念，用户可能只想与朋友分享信息。这

意味着人们可以入侵公有云，并提取有关 Facebook 用户的信息。这是一个目前存在争议的话题。

知识管理：公司可以将其知识管理业务外包给云服务提供商。也就是说，为了加强组织业务策略，所有相关信息都存储在云端。使用比如数据挖掘之类的技术来分析信息，以便向高级管理层通报需要进行的任何更改。这里需再次说明，机密性是一个问题。一个组织不会想要将其秘密和知识产权公开。因此，组织可能会使用私有云进行工作。

总而言之，许多使用云的应用程序需要安全，需要确保个人的隐私，并在许多情况下确保数据的机密性。因此，私有云似乎最适合这样的各种应用。然而，私有云的托管和实施是昂贵的。人们希望尽可能地利用公有云。因此，为了使用公有云，安全性至关重要。另一个选择是使用混合云。也就是说，对非敏感应用程序使用公有云，而对敏感应用程序使用私有云。

11.4 总结和展望

在本章中，我们讨论了专用云和云应用。特别是我们讨论了移动云和专用云以及防御、金融、社交网络和知识管理等应用。初步研究了第 7 章中讨论过的一些专用的 Web 服务，并将其应用于提供云应用服务。我们的云实验系统以及应用将在本书的第四部分、第六部分和第七部分进行讨论，包括基于云的恶意软件检测、基于云的内部威胁检测和基于云的社交网络。我们还将讨论云中的查询处理的安全。

需要注意的是，第 10 章和第 11 章讨论了数据管理，而框架的应用层已经在第 9 章讨论过了。我们简要讨论了虚拟化、存储和网络层框架的其他层面。在第 12 章中，我们将论述已经出现的各种云产品。作为本书第三部分主题的安全问题，还将出现在本书的第五部分进行必要的论述。

参考文献

[ANDROID] Android operating system, http://google.about.com/od/socialtoolsfromgoogle/p/android_what_is.htm
[AWS] i-Phone application hosting, http://aws.amazon.com/iphone-application-hosting/
[IBM] IBM developerWorks http://www.ibm.com/developerworks/topics/android/
[MCC] Mobile Cloud Computing, http://en.wikipedia.org/wiki/Mobile_cloud_computing
[ZHU11] Zhu, W., C. Luo, J. Wang, and S. Li, Multimedia cloud computing, *IEEE Signal Processing Magazine,* 28(3), 59–69, May 2011.

第 12 章　云服务提供商、产品和框架

12.1　概述

大量云服务提供商从 2007 年左右开始出现，其中值得注意的有亚马逊网络服务、Google 应用引擎、微软的 Azure、Proofpoint、Rightscale、Salesforce. com、Sun 的开放云平台和 Workday（见本章参考文献［AWS］、［GOOG］、［WIND］）。此外，许多云产品不仅来自亚马逊、谷歌和微软等服务提供商，还可能出自比如 Oracle、VMware 和 IBM（见本章参考文献［IBM］、［ORAC1］、［ORAC2］、［VM］）等提供商。同时，也出现了云框架结构，包括 Storm、Hadoop / Map/Reduce 和 Hive（见本章参考文献［HIVE］、［HADOOP］、［STORM］）。一些研究原型，如 XEN 虚拟机管理程序也在商业化（见本章参考文献［BARH03］）。

对产品进行分组是一项艰巨的任务，因为服务提供商本身也是市场产品。例如，亚马逊的弹性计算机云（Elastic Computer Cloud，EC2）与亚马逊简单存储服务（Amazon Simple Storage Service，AmazonS3）、亚马逊关系数据库服务（Amazon Relational Database Service，Amazon RDS）、亚马逊 SimpleDB 和亚马逊简单队列服务（Amazon Simple Queue Service，Amazon SQS）协同工作，提供计算、查询处理和跨应用程序存储的解决方案。因此，我们将 Amazon、Microsoft 和 Google 等服务提供商的产品，与 Oracle Enterprise Edition、IBM SmartCloud、VMware 和 XEN 等产品，以及 Hadoop、Map/Reduce、Storm 和 Hive 等框架统一起来，并将其描述为产品。应该说明的是，对于所论述的内容我们只选择了最熟悉的服务提供商、产品和框架，或是已经在工作中测试过的。而描述所有服务提供商、产品和框架超出了本书的范围。

本章的结构如下：在 12.2 节中，我们将描述各种云服务提供商：亚马逊、微软和谷歌、Oracle、IBM、VMware 和 Citrix（即 XEN）等生产商，以及比如 Hadoop、Map/Reduce、Storm 和 Hive 等架构。在 12.3 节对本章进行总结。图 12.1 说明了本章讲述的概念。

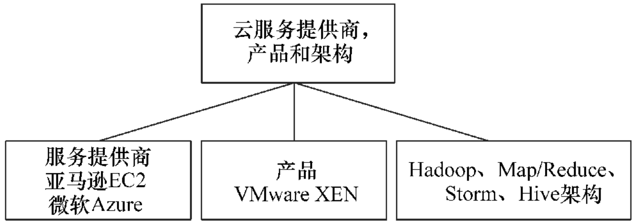


图 12.1　云服务供应商，产品，和架构

12.2　云服务提供商、产品和框架

12.2.1　云服务提供商

本节讨论如图 12.2 所示的云服务提供商。

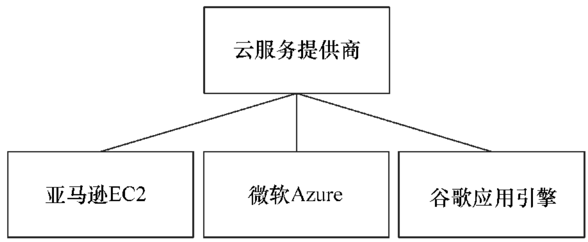


图 12.2　云服务提供商

亚马逊弹性计算云（Amazon EC2）在本章参考文献［AWS］中有所描述，它是一种在云中提供可调整计算能力大小的 Web 服务，其 Web 服务接口中允许获取和配置计算资源容量。它可以让用户完全控制计算资源，它也可以让用户只为其所用的资源容量支付费用。最后，它为开发人员提供开发故障恢复应用程序的工具。Amazon EC2 提供虚拟计算环境，允许用户访问 Web 服务接口，以启动具有各种操作系统的实例，使用自定义应用程序环境加载它们，并管理网络的访问权限和运行影像。

EC2 这个云提供的服务如下：

弹性：Amazon EC2 可以在数分钟内提升或降低计算资源容量。

控制：用户控制他/她的实例。

灵活性：用户可以选择多种实例类型、操作系统和软件包。

与亚马逊网服务集成（AWS）：Amazon EC2 与亚马逊简单存储服务（Amazon S3）、亚马逊关系数据库服务（Amazon RDS）、亚马逊 SimpleDB 和亚马逊简单队列服务（Amazon SQS）合作，来提供计算、查询处理和跨应用程序存储的解决方案。

可靠性：Amazon EC2 提供了一个高度可靠性的环境，可以对替换实例进行调试。

安全性：Amazon EC2 具有保护用户计算机资源的机制。

Amazon EC2 的功能包括：

■ 亚马逊弹性块存储：亚马逊弹性块存储（Elastic Block Store，EBS）为 Amazon EC2 实例提供持久存储。Amazon EBS 提供的是一种与实例无关的存储，它独立于实例的生命周期。

■ 多定位：Amazon EC2 提供在多个地点放置实例的功能。Amazon EC2 位置由区域和可用区组成。

■ 弹性 IP 地址：弹性 IP 地址是专为动态云计算环境设计的静态 IP 地址。弹性 IP 地址与账户相关联，而不是特定实例，并且账户拥有者控制该地址，直到它被明确释放。

■ 亚马逊虚拟私有云：亚马逊 VPC 是现有 IT 基础架构与 AWS 云之间安全无缝的桥梁。Amazon VPC 使企业能够通过虚拟专用网（Virtual Private Network，VPN）连接，

将现有基础设施连接到一组独立的 AWS 计算资源，并扩展其现有的管理功能，如安全服务、防火墙和入侵检测系统，来包含他们的 AWS 资源。

■ 亚马逊云监控：亚马逊云监控是一种 Web 服务，用于监控 AWS 云资源和应用程序。它使得用户可以看到资源利用率、运营绩效和总体需求模式。

■ 自动缩放：自动缩放功能允许用户根据所定义的条件自动缩放他/她的 Amazon EC2 计算资源容量。

■ 弹性负载平衡：弹性负载平衡自动将传入的应用程序流量分配到多个 Amazon EC2 实例。它可以在应用程序中实现容错。

■ 高性能计算（High－Performance Computing，HPC）集群：具有复杂计算工作负载的客户机（如紧耦合并行进程），可以获得与定制基础架构相同的高计算和网络性能。

■ VM 导入/导出：VM 导入/导出使得用户可以轻松地将 VM 映像从现有环境导入到 Amazon EC2 实例中，并随时将其导出。

■ AWS 市场：AWS 市场是一个在线商店，可帮助客户查找、购买和快速部署在 AWS 上运行的软件。

12.2.1.1　Windows Azure

如本章参考文献［WIND］所述，Windows Azure 是一个开放灵活的云平台，可使客户能够在全球微软管理的数据中心网络中快速构建、部署和管理应用程序。人们可以使用任何语言、工具或框架构建应用程序，并可以将公有云应用程序与现有的 IT 环境相集成。本章参考文献［WIND］中已经说明，Windows Azure 每月提供 99.95% 的 SLA，可以在不关注基础设施的情况下构建和运行高可用性应用程序。它提供自动化操作系统和服务补丁、内置网络负载平衡和应对硬件故障的复原功能。它支持可以在不停机的情况下升级应用程序的部署模型。

Windows Azure 可以使用任何语言、框架或工具来构建应用程序。由于使用开放的 REST 协议，其功能和服务是公开的。Windows Azure 客户端库支持多种编程语言，并通过开源许可证发布。在本章参考文献［WIND］中也有说明，Windows Azure 可以轻松地将应用程序扩展到任何大小。它是一个自动化的自助服务平台，可以在几分钟内为客户提供资源。客户也可以增加或缩减资源使用量，并且仅支付其所使用的资源。

通过 Windows Azure，数据可以利用关系 SQL 数据库、NoSQL 的表存储和非结构化的 blob 类型进行存储，也可以选择使用 Hadoop 和商业智能服务进行数据挖掘。Windows Azure 的分布式缓存减少了客户的等待时间。此外，每个计算实例都是将客户与其他客户隔离的虚拟机。一旦将计算资源分配给客户的应用程序，Windows Azure 将自动处理网络工作量平衡和故障切换，以提供持续的可用性。

Windows Azure 的组件包括 SQL Azure 和 Windows Azure 存储。如本章参考文献［WIND］中所述，SQL Azure 是一种基于 SQL 服务器技术的高度可用的，且可缩放的云数据库服务。利用 SQL Azure，开发人员无需安装、设置或管理任何数据库。本身内置高度可用性和容错性，无需进行物理管理。SQL Azure 是一项由微软运营的托管服务，

每月提供 99.9% 的 SLA。SQL Azure 提供了一个功能完备的关系数据库，并实现了多种不同类型的应用程序，一些常见方案包括业务应用程序、混合应用程序和商业智能。如本章参考文献［WIND］中所述，Windows Azure 存储提供安全、可扩展和易于访问的存储服务，且能保持高可用性和持久性。存储服务支持多种存储格式，如结构化和非结构化数据、NoSQL 数据库和队列。存储是一种托管服务，每月提供 99.9% 的 SLA。Azure 的其他组件包括用于网络、商业智能、内容传递和安全性。更多细节请参见本章参考文献［MATH09］。

12.2.1.2　谷歌应用引擎

如本章参考文献［GOOG］所述，谷歌应用引擎可让客户在 Google 的基础设施上使用网络应用。应用程序的应用引擎易于构建、维护，易于随着流量和数据存储需求的增长而扩展。使用应用引擎时，不需要服务器去维持，而且用户可以上传他/她的应用程序，并在云上运行。谷歌应用引擎支持使用多种编程语言编写的应用程序。可以使用谷歌应用引擎的 Java 运行环境，也可以使用标准 Java 技术构建应用程序，包括 JVM、Java servlet 和 Java 编程语言。应用引擎还具有两个专用的 Python 运行环境。使用应用引擎，客户只需要为其所使用的部分付费。

如本章参考文献［GOOG］所述，应用引擎包含以下功能：

■ 动态 Web 服务，全面支持通用的网络技术；

■ 查询、排序和事务的持久性存储；

■ 自动扩展和负载平衡；

■ 用于验证用户和使用 Google 账户发送电子邮件的 API；

■ 全功能的本地开发环境，它可在您的计算机上模拟谷歌应用引擎；

■ 用于在 Web 请求的范围之外执行作业的任务队列；

■ 在指定时间和定期间隔触发事件的计划任务。

应用引擎环境提供了一系列数据存储选项，包括以下内容：

■ 应用引擎 Datastore 提供了一个 NoSQL 无模式对象数据存储区，其中包含查询引擎和原子事务。

■ 谷歌云 SQL 为基于 MySQL 的应用引擎应用程序提供了一个关系型 SQL 数据库服务。

■ 谷歌云存储可以为对象和文件提供存储服务，这些对象和文件的大小可以达到 TB 级别且能够由 Python 和 Java 应用程序访问。

有关应用引擎的更多详细信息，请参见本章参考文献［MATH09］。

12.2.2　云产品

本节讨论如图 12.3 所示的云产品。

12.2.2.1　Oracle 企业管理器

在本章参考文献［ORAC1］中指出，Oracle 企业管理器是 Oracle 的综合企业 IT 管理生产线，并提供了一个云生命周期管理解决方案。它使客户能够建立、管理和支持企

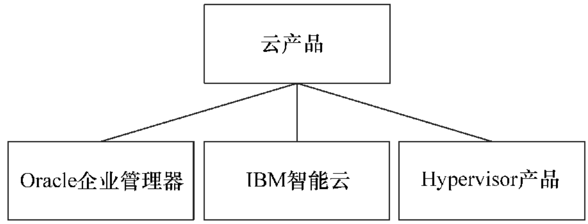


图 12.3　云产品

业云。本章参考文献［ORAC1］中还表示，Oracle 企业管理器是 Oracle 首屈一指的云管理解决方案。它提供自助服务配置，来权衡集中式、基于策略的资源管理、集成的扣款和容量规划以及从应用程序到磁盘的物理和虚拟环境的可见性。实际上，企业管理器为企业中的用户提供数据库即服务。

如本章参考文献［ORAC2］所述，用于 Oracle 数据库的 Oracle 云管理包提供了跨越整个数据库云生命周期的相关功能。它允许云管理员识别集合资源，配置基于角色的访问，定义服务目录以及相关的扣费计划。它允许云用户请求数据库服务，并按需消费。它还允许用户扩展和缩小平台以适应工作负载的变化。最后，它让双方了解所提供服务的成本，并建立了消耗资源的问责制。

在本章参考文献［ORAC1］中指出，企业管理器已经将它与一个即买即用的自助服务门户相关联，允许开发人员、测试人员、数据库管理员和其他自助服务用户登录并请求新的单实例（Single Instance，SI）和实际应用集群（Real Application Clusters，RAC）数据库，以及执行生命周期操作，如启动/停止、状态和健康监控。还可以在 OracleVM 虚拟化服务器基础架构上，部署包含数据库的虚拟程序集。该门户提供对服务目录的访问，目录中列出了用于标准化数据库配置和各种已发布的版本服务模板。用户可以查看他们过去和未完成的请求、资源配额、当前利用率，以及他们所拥有数据库的扣费信息。该门户还允许用户每天自动备份数据库或进行按需备份。用户可以将数据库恢复到任何备份的状态。自助服务门户是用户进入云的接口视图。

Oracle 还发布了一些云产品，来增补企业管理器，其中包括 Oracle 文件系统。此外，Oracle 还宣布将推出一套基于云计算的产品，包括客户关系管理云。

12.2.2.2　IBM 智能云

如本章参考文献［IBM］所述，智能云是 IBM 对云计算的愿景。IBM 智能云基础架构产品包括用于构建私有云和混合云计算环境的服务器、存储和虚拟化组件。它提供基础设施即服务解决方案，可以快速部署云。它还提供自动配置、并行可扩展性、集成容错能力，以及更高级云功能的基础。

IBM 智能云配置如下：

- ■ 一种低成本，易于使用的私有云解决方案，可在数小时内部署。
- ■ 可靠、不间断的操作，能够自动容忍和恢复软件和硬件故障。
- ■ 快速的可扩展性，可以快速部署数百台虚拟机来满足业务增长。
- ■ 低触摸式基础架构可帮助减少错误，增强安全性和合规性，并提高管理员的工作效率。

IBM 智能云监控私有云基础架构的运行状况和性能，包括物理和虚拟化组件的环

境。根据需求，该解决方案还可以提供评估当前容量和模型扩展所需的工具。与 SmartCloud 相关的是一套用于集成多云、数据管理和安全性的工具。例如，IBM 的 DB2. 9 是用于管理关系数据和 XML 数据的数据库服务器，适用于云部署。

12. 2. 2. 3 虚拟机管理程序产品

自 20 世纪 90 年代后期以来，已经有几款虚拟机管理程序的产品面世。早期的虚拟机管理程序的产品之一就是 VMware。它一开始作为斯坦福大学研究项目，然后被商业化。实际上，VMware 开发了云端的一个早期虚拟化软件。如本章参考文献［VM］所述，VMware 软件为访客操作系统提供了一套完全虚拟化的硬件。VMware 软件可以虚拟化视频适配器、网络适配器和硬盘适配器等硬件。主机为访客 USB、串行和并行设备提供传递驱动程序。这样一来，VMware 中的虚拟机就可在计算机之间进行移植，因为每个主机看起来均与访客几乎相同。VMware 提供桌面虚拟化、服务器虚拟化、云虚拟化和应用程序。其桌面产品套件包括 VMwareWorkstation 和 VMwareFusion。其服务器产品包括 VMwareESX 和 VMotion。其云产品包括 VMwarevCloud。其应用产品包括 VMware vFabric tc 服务器。

另一个著名的管理程序产品是 XEN。像 VMware 一样，它是作为英国剑桥大学的研究项目而开始的，并通过 Citrix 进行销售。XEN Hypervisor 运行在硬件之上，并捕获虚拟机的所有请求以访问硬件。域 0（Dom0）是用于管理其他虚拟机的 Linux 修改版本。域 U（DomU）是 XEN 中的用户域。DomU 是所有不受信任的客户操作系统所在的地方。DomU 分为两部分：Para 虚拟化域（Para - Virtualized Domains，PV）和硬件辅助虚拟机（Hardware Assisted Virtualized Machine，HVM）域。Para 虚拟化域是一个已修改的操作系统，要意识到它是一个虚拟机，且可以实现近乎原生的性能。HVM 域是一个虚拟机且运行尚未修改为工作在 Dom0 的操作系统。PV 被授予对存储器的只读访问，并且任何更新操作由管理程序控制。给出一个影子页表，因为它们不知道如何使用不连续的物理地址空间。I/O（输入/输出）管理由 Dom0 控制。PV 与 Dom0 共享内存，通过它们可以传递消息。

VMware 和 XEN 都有几篇白皮书和文章。详细信息请参见本章参考文献［MATH09］。我们将在第 17 章讨论这些管理程序的安全细节。

12. 2. 3 云框架

本节讨论云服务提供商，如图 12. 4 所示。我们在实验工作中利用了这些框架，实验系统将在第四部分讨论。

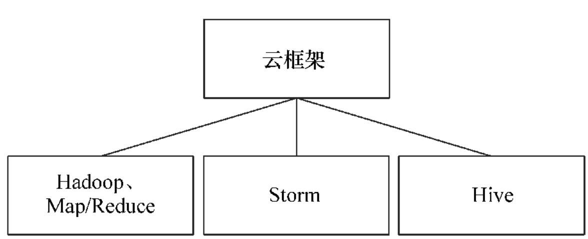


图 12. 4　云框架

12. 2. 3. 1 Hadoop、Map/Reduce 框架

如本章参考文献［AWS］所述，Apache Hadoop 是一个软件框架，可以在免费许可证下支持分布式应用程序。它还可以使应用程序在许多机器上工作，并生成大量的数

据。Hadoop 源于 Google 的 Map/Reduce 和 GoogleFileSystem（GFS）系列文件内容，它是用 Java 编写的。

Hadoop 由 Hadoop Common 组成，其可以访问 Hadoop 支持的文件系统。为了有效地调度工作，每个 Hadoop 兼容文件系统都应提供位置侦测：工作节点所在机架的名称。Hadoop 应用程序还可以使用此信息，在数据所在的节点上运行工作，并且在同一个机架/交换机上运行该功能，从而减少主干网流量。HDFS 在复制数据时使用此功能，尝试将数据的不同副本保留在不同的机架上。其目标是减少机架停电或开关故障的影响，以便即使发生这些事件，数据仍然可以读取。

一个小的 Hadoop 集群将包含一个主节点和多个工作节点。主节点由 JobTracker（作业跟踪器）、TaskTracker（任务跟踪器）、NameNode（名称节点）和 DataNode（数据节点）组成。从属节点或工作节点既充当数据节点，也充当任务跟踪器。虽然它们可能是仅有数据的工作节点和仅用于计算的工作节点，但这些通常仅用于非标准应用。在较大的集群中，专用名称节点服务器进行管理 HDFS，以托管文件系统索引，并且二次名称节点可以生成名称节点内存结构快照。这样就可以防止文件系统损坏并减少数据丢失。

HDFS 是用 Java 编写的用于 Hadoop 框架的分布式、可扩展和可移植的文件系统。Hadoop 实例中的每个节点通常都具有单个数据节点，及一组数据节点形成的 HDFS 集群。位于文件系统之上的 Map/Reduce 引擎，由一个作业跟踪器组成，客户端应用程序向其提交 Map/Reduce 作业。作业跟踪器将工作推送到可用任务跟踪器节点的集群中，努力使作业尽可能靠近数据。使用机架感知文件系统，作业跟踪器可以知道哪个节点包含数据，哪些其他机器在附近。如果作业无法托管在数据所在的实际节点上，则优先考虑同一机架中的节点。这减少了主干网上的网络流量。如果任务跟踪器失败或超时，该部分作业将重新安排。

我们在所有的云中都使用了 Hadoop/Map/Reduce 框架，这些在后面的章节中进行讨论。也就是说，我们的许多原型，包括云数据管理器和社交网络都托管在 Hadoop/Map/Reduce 框架上。

12.2.3.2 Storm

Storm 是一种开放源码的分布式实时计算系统。Storm 类似于 Hadoop，它为用户提供了实时执行计算的通用框架，就像 Hadoop 为用户提供执行批量处理操作的通用框架一样。Storm 提供以下关键属性：①支持广泛的用例，如流处理和连续计算；②Storm 是可扩展的，具有每秒能处理大量信息的能力；③Storm 保证每个信息都将被处理，从而确保没有数据遗落；④Storm 集群易于管理，非常强大；⑤Storm 通过自动重新分配在执行过程中失败的任务来确保容错性；⑥Storm 的组件和编程语言无关，因此几乎可以被任何人使用。

Storm 集群由两种节点组成：主节点和工作节点。主节点运行一个称为“Nimbus”的守护进程，负责在集群周围分发代码，将任务分配给机器并监控故障。每个工作节点运行一个名为“Supervisor”的守护进程，它监听 Nimbus 为其分配的工作，并启动/停

止工作进程以完成此工作。Nimbus 和 Supervisor 之间的协调是通过 Zookeeper 集群完成的。

Storm 使用“拓扑”的概念进行实时计算。Storm 拓扑类似于 Map/Reduce 作业，但是，一个关键的区别是，Map/Reduce 作业最终会完成，而拓扑中会永久运行或直到被终止。拓扑是由 spout 和 bolt 构成的有向图，spout 与 bolt 通过流分组连接在一起。值得注意的是，流是 Storm 中的核心抽象，并且表示使用分布式集群以并行方式创建和处理的无限制的元组序列。spout 作为拓扑的流源，通常从外部源读取元组，并将其发布到拓扑中。bolt 用于在拓扑中执行所需的处理，例如启动与数据库的连接，执行比如过滤、聚合和连接等操作。流分组确定输入到一个特定 bolt 的流如何在该 bolt 的任务之间进行分离。最后，Storm 保证每个 spout 元组都将被拓扑完全处理，并且还提供了几种配置来定制 Nimbus、Supervisor 和运行拓扑的行为。

12.2.3.3 Hive

Hive 是 Apache 的开源系统。如本章参考文献［HIVE］所述，Hive 是一个用于 Hadoop 的数据仓库系统，可帮助您轻松进行数据汇总、点对点查询以及对存储在 Hadoop 兼容文件系统中的大型数据集进行分析。Hive 提供了一种机制来将结构投影到这些数据上，并使用类似 SQL 的语言 HiveQL 来查询数据。同时，这种语言还允许传统的 Map/Reduce 程序员在不方便或低效时插入自己的定制映射器和规约器，以在 HiveQL 中表达这种逻辑。也就是说，Hive 支持以类似 SQL 的语言进行数据查询、转换为 Map/Reduce 任务，然后由 Hadoop 框架执行。使用 HDFS 作为数据存储，Hive 继承了 Hadoop 的所有容错能力和可伸缩性，以及处理巨大数据集的能力。

如本章参考文献［HIVE］所述，Hive 查询语言和 SQL 查询之间的主要区别是 Hive 查询在 Hadoop 集群（而不是为了大型数据集而使用昂贵的硬件构建平台）上执行。这样就允许 Hive 可以扩展，以处理大量数据集。Hive 查询的内部执行是通过一系列自动生成的 Map/Reduce 作业来完成的。

Hive 是执行 Hive 查询的基础。它包括三个主要部分：Hadoop 集群、元数据存储、仓库主管。Hadoop 集群是存储大数据集和执行处理廉价商品计算机的集群。元数据存储是保存大型数据集结构描述的位置。仓库目录是 Hive 用于存储/缓存工作文件的存储位置。

一种引人关注 Hive 的实现方法，来自于 Facebook 的开发人员。在一篇文章（见本章参考文献［THUS09］）中，Facebook 开发人员已经将 Hive 描述为构建在 Hadoop 之上的开源数据仓库解决方案。他们表示，在 Facebook 中，Hive 仓库包含数以万计的桌面和超过 700 TB 的数据，并被广泛用于 200 个用户的每月报告和点对点分析。我们在实现的云查询处理中也广泛使用了 Hive，将在第 22 章讨论具体的实现过程。

12.3 总结和展望

在本章中，我们讨论了各种云服务提供商、产品和框架。这些包括亚马逊、微软和

谷歌等服务提供商，Oracle、IBM、VMware 和 XEN 等产品以及 Hadoop、Map/Reduce、Storm 和 Hive 等框架。需要注意的是，我们选择的这些都是我们熟悉的服务提供商、产品和框架，并已将其用于我们的工作中。还有更多优秀的云服务提供商、产品和框架，但是如果对它们全部讨论超出了本书的范围。本章参考文献［MATH09］也给出了一个概述。

第三部分主要介绍了有关云计算中的一些关键概念，包括部署和服务模型、功能和应用。我们将在第四部分讨论如何使用各种产品、框架和服务的一些细节。云安全问题以及我们的云计算安全实验系统将成为第五到九部分的主题。特别地，我们将讨论各种云服务、产品和框架的安全性。

参考文献

[AWS] Amazon Web Services: http://www.aws.amazon.com/ec2/

[BARH03] Barham, P., B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, Xen and the art of virtualization. *Proceedings of the 19th ACM Symposium on Operating Systems Principles 2003 (SOSP 2003)*, Bolton Landing, NY, 164–177.

[GOOG] Google Apps Engine, http://www.oracle.com/technetwork/oem/pdf/512029.pdf

[HADOOP] Hadoop, http://hadoop.apache.org/

[HIVE] The HIVE system, http://hive.apache.org/

[IBM] IBM cloud computing, http://www.ibm.com/cloud-computing/us/en/

[MATH09] Mather, T., S. Kumaraswamy, S. Latif, *Cloud Security and Privacy*, O'Reilly, Sebastopol, CA, 2009.

[ORAC1] Oracle enterprise manager, http://www.oracle.com/us/products/enterprise-manager/index.html

[ORAC2] Oracle cloud management pack for Oracle database, http://www.oracle.com/technetwork/oem/pdf/512029.pdf

[STORM] Twitter storm: Open source real-time Hadoop, http://www.infoq.com/news/2011/09/twitter-storm-real-time-hadoop

[THUS09] Thushoo, A., Hive—A petabyte scale data warehouse using Hadoop, http://www.facebook.com/note.php?note—id=89508453919

[VM] VMware virtualization overview, http: //www. vmware.com/pdf/virtualization.pdf

[WIND] Windows Azure, http://www.windowsazure.com/en-us/develop/overview/

第三部分总结

在第二部分我们讨论了云计算核心的 Web 服务，第三部分我们概述了云计算概念、技术和产品。我们还展示了如何利用第二部分关于 Web 服务、语义 Web 服务和专业 Web 服务的概念为消费者提供云服务。

第 8 章介绍了云的概念，并讨论了虚拟化的各个方面。特别是讨论了硬件虚拟化、操作系统虚拟化、网络虚拟化和数据虚拟化等方面。还论述了云的各种服务模式和部署模式。

第 9 章讨论了云计算功能的各个方面。首先，我们提出了一个云计算框架。该框架由网络层、虚拟化层、存储层、数据管理层和应用层组成。其次，阐述了每层的功能。最后，我们讨论了云的策略管理、备份和恢复问题。

第 10 章论述数据管理系统的各个方面，并研究了云的影响。特别是讨论了云查询处理、云事务管理和云存储管理等方面。还讨论了云中的信息管理、知识管理和活动管理。

第 11 章讨论了两个主题：专用云和云应用，特别是我们讨论了移动云和专用云以及云应用，如国防、金融、社交网络和知识管理。

第 12 章讨论了各种云服务提供商、产品和框架。这些包括亚马逊、微软和谷歌等服务提供商，Oracle、IBM、VMware 和 XEN 等产品，以及 Hadoop、Map/Reduce、Storm 和 Hive 等框架。

第四部分　云计算实验系统

第四部分简介

既然我们已经介绍了关于云计算的基本情况，接下来我们将介绍一些已经开发的云计算实验系统。这将使读者更好地了解如何利用分布式计算、Map/Reduce 和 Storm 等这些在第三部分已讨论过的技术。

第四部分由三个章节组成：第 13、14 和 15 章。第 13 章讨论关于语义 Web 云查询处理原型系统。第 14 章讨论基于云服务的社交网络托管工作。第 15 章讨论我们已经开发的多个原型系统，包括面向社会网络的云计算、面向语义 Web 数据存储的云计算、基于本体的查询处理的云计算。

第 13 章　云查询处理实验系统

13.1　概述

如前所述，云计算是一个在 IT 以及数据处理领域中的新兴计算模式。企业利用云计算服务进行数据源的维护，从而产生显著的经济效益。他们也能利用云服务实现数据的存储与远程数据的访问。随着云计算的普及，云计算服务提供商将面临越来越大的挑战。他们在提供高效的信息检索功能的同时，还必须维护大量的异构数据。因此，云计算解决方案的重心是可扩展性和查询效率。语义 Web 技术正在发展，力图用标准化的方式来呈现数据，这样的数据既可以被人理解，同时也可以被机器检索。从以往经验上看，网页以 HTML（Hypertext Markup Language，超文本标记语言）文件展现出来，而这些文件不适合推理。机器将这些 HTML 文件视为一系列关键字。研究人员正在开发标准化的语义 Web 技术，以解决这些不足之处。最突出的标准是 RDF（见本章参考文献［W3b］）协议、SPARQL（SPARQL 协议和 RDF 查询语言）协议和 RDF 查询语言（见本章参考文献［W3c］）（SPARQL）协议。RDF 是存储和表示数据的标准，SPARQL 是从 RDF 数据库检索数据的查询语句。云计算系统可以利用这些语义 Web 技术的力量，为用户提供面向数据密集型应用程序的有效存储数据和检索数据的功能。

语义 Web 技术对云数据的维护尤为重要。这些技术能够以标准化的方式来指定和查询异构数据。而且，通过在 Web 本体语言（Web Ontology Language，OWL），可以指定本体、不同的模式、类、数据类型和关系，而不会影响标准 RDF/SPARQL 接口。反过来，云计算的解决方法可以很好地应用于语义 Web 社区。语义 Web 数据集正呈指数级增长。因此，与其他领域相比，在 Web 领域中，可伸缩性是最重要的。同时，快速的响应在网络社区中也至关重要，这是因为有大量的客户（即用户）访问大量的 Web 数据。我们相信，云计算模式提供了一个可以达到这两个目标的解决方案。

现有的商业工具和技术在云计算环境中并没有得到很好的扩展。最近，研究人员开始关注这些问题。他们建议从零开始建立系统。在本章参考文献［WANG10］中，研究人员提出了一种新的分布式数据库（见本章参考文献［COMP］）索引方案，可以用来作为一个云系统。当涉及如 RDF 的语义 Web 数据时，我们就会面临着类似的挑战。尽管需要存储和检索大量的数据，但是随着存储成本变得越来越低，为处理数十亿个 RDF 三元组，我们需要开发 TB 级甚至 PB 级磁盘空间的系统将不再遥远。研究人员正致力于此需求的开发（见本章参考文献［NEWM08］、［ROHL07］），同时也组织了各类比赛以鼓励研究人员建立有效的知识库（见本章参考文献［CHAL］）。目前，语义 Web 技术只有几个框架［例如，RDF-3X（见本章参考文献［NEUM08］），Jena（见本章参

考文献［CARR04］），Sesame（见本章参考文献［OPEN］），BigOWLIM（见本章参考文献［KIRY05］），而这些框架对大型 RDF 图表有所局限。因此，存储大量的 RDF 三元组并且高效地查询它们是一个具有挑战性的重要问题。

在这一章中，我们讨论具有云计算功能和管理大量 RDF 三元组功能的查询处理系统。本章结构如下：在 13.2 节中，讨论我们的方法。在 13.3 节中，讨论相关的工作。在 13.4 节中，讨论系统架构。在 13.5 节中，我们讨论如何响应 SPARQL 查询。在 13.6 节中，给出实验结果。最后，在 13.7 节中，我们得出一些结论，并讨论一些我们已经确定且在未来需要进行改进的领域。本章内容如图 13.1 所示。更详细的讨论的概念、体系结构和实验在本章参考文献［HUSA11a］和［HUSA11b］中提供。

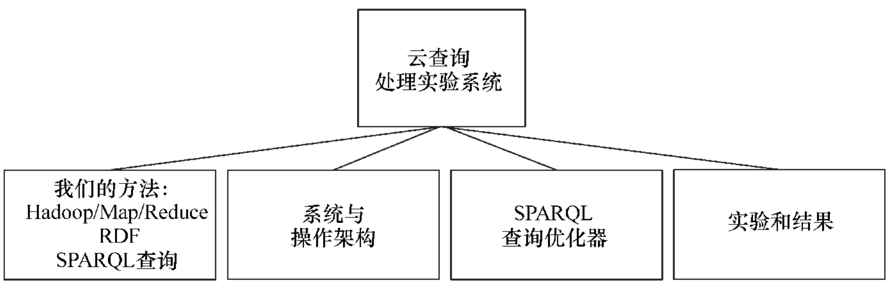


图 13.1　云查询处理［Husain，M. F.，J. P. McGlothlin，M. M. Masud，L. R. Khan 和 B. M. Thuraisingham，Heuristics－based query processing for large RDF graphs using cloud computing，IEEE Transactions on Knowledge and Data Engineering，23（9）：1312－1327. ©（2011）IEEE.］

13.2　我们的方法

我们可以构建一个分布式系统，以克服当前语义 Web 框架的可扩展性和性能问题。数据库的分布式存储，就是提供这种可扩展性的解决方案。但是，迄今为止并没有发现用于存储和管理 RDF 数据的分布式资料库。研究人员最近才开始探索构建这样的分布式系统所必须解决的问题和技术的解决方案。目前，有一个充满前景的调查研究路线，其所涉及利用现成的分布式数据库系统或关系数据库。这样的数据库系统可以使用关系模式来存储 RDF 数据。首先要将它们转换为 SQL 可以应答的 SPARQL 查询（见本章参考文献［CHEB07］、［CHON05］、［CYGA05］）。为了达到这个目的，需要研究最佳关系模式（见本章参考文献［ABAD07］）。这种系统的主要缺点是，它们是针对关系数据进行优化的。对 RDF 数据的处理不是很好，特别是因为 RDF 数据是三元组的集合（见本章参考文献［W3a］）（称为主题、谓词和对象的三个组件的有序元组），它们形成大的有向图。在 SPARQL 查询中，任何数量的三元组模式（TP）（见本章参考文献［W3e］）都可以加入单个变量（见本章参考文献［W3d］），这使得关系数据库查询计划变得复杂。性能和可扩展性仍然是一个具有挑战性的问题，因为这些系统是针对关系数据模式和事务性数据库的使用而优化的。

另一种方法是从头开始构建一个分布式 RDF 系统。在这里，会设计和优化一个特定的应用于 RDF 数据的系统。然而，这种做法是高度定制的，将很难适应新的标准。因为我们并不是从头开始，而是用一个通用的分布式存储系统构建了一个解决方案，它利用了云计算平台。然后，我们专门为满足语义 Web 数据的需要量身定做了系统和架构。最后，我们构建了一个使用这种存储设备的语义 Web 存储库。

Hadoop（见本章参考文献［HADOa］）是一个分布式文件系统，可以通过复制保存文件。它是构建存储系统的理想选择。Hadoop 具有高容错性和极高的可靠性。此外，它还包含了 Map/Reduce（见本章参考文献［DEAN04］）编程模型，这是一种适合于并行处理大量数据的函数编程模型。通过将数据分割成多个独立的区块，Map/Reduce 进程针对这些区块运行，使并行化变得更加简单。此外，Map/Reduce 编程模型有助于简化加入多个三元组模式的任务。

在本章中，我们将描述一种在 Hadoop 中存储 RDF 数据的模式，并详细介绍一种针对处理这些数据查询的解决方案。在预处理阶段，我们处理 RDF 数据并在分布式文件系统中填充文件。该过程包括分区和组织数据文件并执行字典编码。然后，将详细介绍一个用于信息检索的查询引擎。我们将精确地指定如何使用 Map/Reduce 编程来满足 SPARQL 查询。具体来说，我们必须确定将要执行的 Hadoop“作业”来解决查询。我们将介绍一种贪婪算法，其可以生成一个查询计划，且用到最少量的 Hadoop 作业。这虽然是一个基于启发式的近似算法，但是我们将证明最坏的情况具有合理的上限。最后，我们将利用两个标准的基准数据集来运行实验。我们提出的数据集范围，从 1 亿到 66 亿以上的三元组，展示了我们的解决方案非常具有可扩展性。也会证明，我们的解决方案胜过处于领导地位的且具有最先进的语义 Web 存储库，在庞大的数据集上使用标准的基准查询。我们所做出的贡献列在下面，如图 13.2 所示。更多细节在本章参考文献［HUSA11a］中给出。

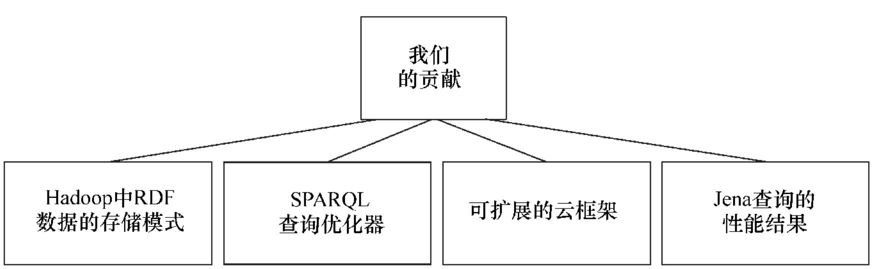


图 13.2　我们的贡献

1）我们设计了一种在 HDFS（见本章参考文献［HADOb］）中存储 RDF 数据的存储方案。

2）我们开发了一种算法，这种算法可以保证提供一个查询计划，其成本由给定 SPARQL 查询中变量总数的日志界定。它使用汇总统计来估计连接选择性，以断开关系。

3）我们构建了一个具有高度可扩展性和容错性的框架，并支持数据密集型查询处理。

4）我们展示了我们的方法对于所有的查询都比 Jena 的好。针对带有大型结果集的复杂查询。它的执行效果要好于 BigWLIM 和 RDF－3X。

13.3 相关工作

Map/Reduce 虽然是一个编程规范，但它正在迅速被研究者们认可。这种技术在处理大量数据的网络社区中越来越受欢迎。它是解决研究人员在云计算中所面临的性能问题的最有前途的技术。在本章参考文献［ABAD09a］中，作者讨论了 Map/Reduce 如何能够满足大多数要求，来构建一个理想的云数据库管理系统。研究人员和企业使用 Map/Reduce 技术进行 Web 索引、搜索和数据挖掘。在本节中，我们将首先说明与 Map/Reduce 有关的研究。紧接着，我们将讨论与语义 Web 相关的工作。

Google 将 Map/Reduce 应用于 Web 索引、数据存储和社交网络（见本章参考文献［CHAN06］）。雅虎在其数据分析任务（见本章参考文献［OLST08］）中广泛使用 Map/Reduce。IBM 已经成功地使用 Map/Reduce 技术来尝试构建一个扩大的向外扩展的搜索框架（见本章参考文献［MORE07］）。在最近的一项工作（见本章参考文献［SISM10］）中，他们报告了如何整合 Hadoop 和 System R。通过将 Hadoop 与并行 DBMS（见本章参考文献［XU10］）集成，Teradata 做了类似的工作。

研究人员使用 Map/Reduce 来扩展分类器，用于挖掘 PB 级别的数据（见本章参考文献［MORE08］）。他们对数据挖掘工作进行了数据分配和分区，并应用了三种数据挖掘算法来测试性能。数据挖掘算法正在以不同的形式重写，以利用 Map/Reduce 技术。在本章参考文献［CHU07］中，研究者重写了众所周知的机器学习算法，利用 Map/Reduce 编程规范来充分发挥多核机器的优势。该技术成功应用的另一个领域是模拟仿真（见本章参考文献［MCNA07］）。在本章参考文献［ABOU09］中，研究者提出了一个有趣的想法，将 Map/Reduce 与现有的关系数据库技术相结合。这些工作与我们的研究不同，因为我们使用 Map/Reduce 的语义 Web 技术。我们的重点是开发一个可扩展的解决方案，用于存储 RDF 数据并使用 SPARQL 查询检索它们。

在语义 Web 领域中，使用 Map/Reduce 技术时，还没有开展太多的工作。我们发现了两个相关的项目：BioMANTA（见本章参考文献［ITEE］）项目和可扩展、高性能、健壮和分布式（切分）项目（the Scalable，High－Performance，Robust，and Distributed，SHPRD）（见本章参考文献［CLOU］）。BioMANTA 建议扩展 RDF 分子（见本章参考文献［DING05］）并实现 基于 Map/Reduce 的分子存储（见本章参考文献［NEWM08］）。使用 Map/Reduce 来回答查询，大约查询了最多四百万个三元组。而我们的工作在以下方面有所不同：第一，查询了十亿个三元组。其次，设计了一个用于改进 RDF 数据的查询执行性能的存储模式。我们根据三元组的谓词和对象的类型将 RDF 三元组存储在文件中。最后，我们还有一个算法来确定一个查询处理计划，其开销是由给定的 SPARQL 查询中的变

量总数的日志界定的。通过此方法的使用，我们可以确定作业的输入文件和应运行的顺序。据我们所知，我们是第一个提出使用 HDFS 中的平面文件来提供 RDF 数据的存储方案，以及用于回答 SPARQL 查询的 Map/Reduce 作业确定性算法。

SHPRD 是使用 Hadoop Cloudera 分布的 RDF 三元组存储系统。此项目显示了初步结果，展示了 Hadoop 提高 RDF 数据集可扩展性的能力。但是，SHPRD 只是存储三元组存储架构中的数据。它当前没有查询计划或重新排序，并且它的查询处理器不会减少 Hadoop 作业的数量。人们对语义 Web 存储库进行了大量的研究，特别强调查询效率和可扩展性。事实上，存在太多这样的存储库可以公平地评估和讨论每一个相应方案的效率和可扩展性。因此，我们将注意语义 Web 存储库，它们是开放的或可供下载的，并且在语义 Web 和数据库社区中得到了良好的认可。

在本章参考文献［ABAD09b］和［ABAD07］中，研究者报告了一个垂直分区的数据库，其用于存储和检索 RDF 数据。它们的解决方案是每个谓词都具有两列表的架构。然后，这种架构在如 CStore（见本章参考文献［STON05］）或 MonetDB（见本章参考文献［BONC06］）的列存储关系数据库之上实现。他们通过对传统型关系数据库的方案来观察并提高性能。我们在 Map/Reduce 框架内的谓词分区中利用了这种技术。但在垂直分割研究中，只使用了小数据库（ <1 亿)。一些论文（见本章参考文献［SIDI08］、［MCGL09］、［WEIS08］）中已经表明，随着数据集的大小的增加，垂直分割方法的性能急剧下降。

Jena（见本章参考文献［CARR04］）是一个语义 Web 框架，用于管理 RDF 数据。对其框架设计而言，它允许集成多个解决方案来实现其持久性。它还通过展开推理的方式支持推断。但是，Jena 仅限于三元组存储架构。换句话说，所有数据都存储在一个三列表中。Jena 对大型数据集的查询性能很差。此外，对数据集的任何更改，都需要对已经推断的三元组全部都要进行重新计算。

BigOWLIM（见本章参考文献［KIRY05］）是最快和最可扩展的语义 Web 框架之一。然而，它不像我们的框架那样可扩展，而且需要非常高端、非常昂贵的机器。它需要昂贵的硬件（大量的主存）来装载大的数据集，而且它的装载时间很长。正如我们的实验所示，当查询中没有绑定对象时，它运行效果并不好。然而，在这种情况下，我们的框架性能表现却不受影响。

RDF－3X（见本章参考文献［NEUM08］）被认为是最快的现有语义 Web 存储库。换句话说，它有最短的查询时间。RDF－3X 使用直方图、汇总统计和查询优化来实现高性能的语义 Web 查询。因此，对于具有绑定对象和聚合查询的功能，RDF－3X 通常能够优于任何其他解决方案。但是，RDF－3X 性能在未绑定查询中呈指数下降。如果选择性因数较低，则连简单连接的查询也会降低。这对于推理查询变得越来越相关，通常要求子查询联合使用未绑定的对象。我们的实验表明，RDF－3X 不仅对此类查询较慢，而且经常中止并无法完成查询。例如，在简单查询“选择所有学生”时。这个 LUBM（利哈伊基准）的查询要求我们选择所有的研究生和本科生，并将结果结合起来。然而，在这个联合中有大量的结果。虽然两个子查询都很容易完成，但联合将在 RDF－

3X 中中止，因为（30000 数据集）有 33 亿三元组的 LUBM。

RDF 知识库（RDF Knowledge Base，RDFKB）（见本章参考文献［MCGL10］）是使用基于比特向量建立关系数据库架构的语义 Web 存储库。RDFKB 比 RDF－3X 或垂直分区实现有更好的查询性能。然而，RDFKB 旨在提供知识库功能，如推理正向链接、不确定性推理和本体对齐。RDFKB 在可扩展性之前优先考虑这些目标。RDFKB 无法加载具有 30 亿个三元组的 LUBM（30000），因此无法与我们的可扩展性解决方案相竞争。

Hexastore（见本章参考文献［WEIS08］）和 BitMat（见本章参考文献［ATRE08］）是为 RDF 索引而优化的主存储器数据结构。这些解决方案在热运行时可能会获得优异的性能，但对于从持久存储的冷运行来说，它们没有优势。此外，它们的可扩展性与可用于主内存 RAM 的数量直接相关。这些产品也不可用于测试和评估。

在我们以前的工作（见本章参考文献［HUSA09］、［HUSA10］）中，我们开发了一种贪婪和穷举搜索算法来生成查询处理计划。然而，穷举搜索算法比较昂贵，而贪婪的搜索算法没有界定，其理论上的复杂性没有定义。在本章中，我们提出了一个有上界的新的贪婪算法。此外，我们还观察到旧的贪婪算法无法生成最优计划的情况。新算法能够在每种情况下获得最优的计划。Join Executer 组件使用 Map/Reduce 框架运行作业，然后它将查询答案从 Hadoop 中转给用户。

13.4　架构

我们的系统架构如图 13.3 所示。它实质上由 SPARQL 查询优化器和在云中实现的 RDF 数据管理器组成。

在图 13.4 中说明了一种可操作的体系架构，它由两个组件组成。图的左侧描述了数据预处理组件，右侧显示了查询应答组件。我们有三个用于数据生成和预处理的子组件。我们使用 N－Triples 转换器组件将 RDF/XML（见本章参考文献［W3f］）转换为 N－Triples（见本章参考文献［W3a］）序列化格式。谓词分割（Predicate Split，PS）组件采用 N－Triples 数据并将其分割成谓词文件。然后，谓词文件被送入谓词对象分割（Predicate Object Split，POS）组件中，后者根据对象的类型将谓词文件分割成较小的文件。下面介绍了这些步骤。

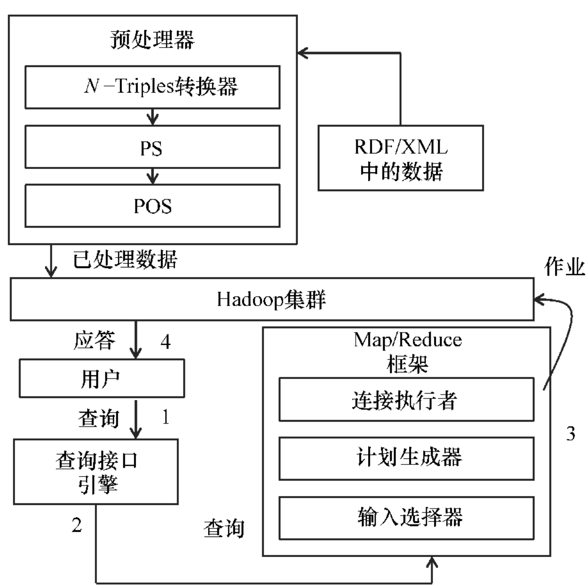


图 13.3　系统架构

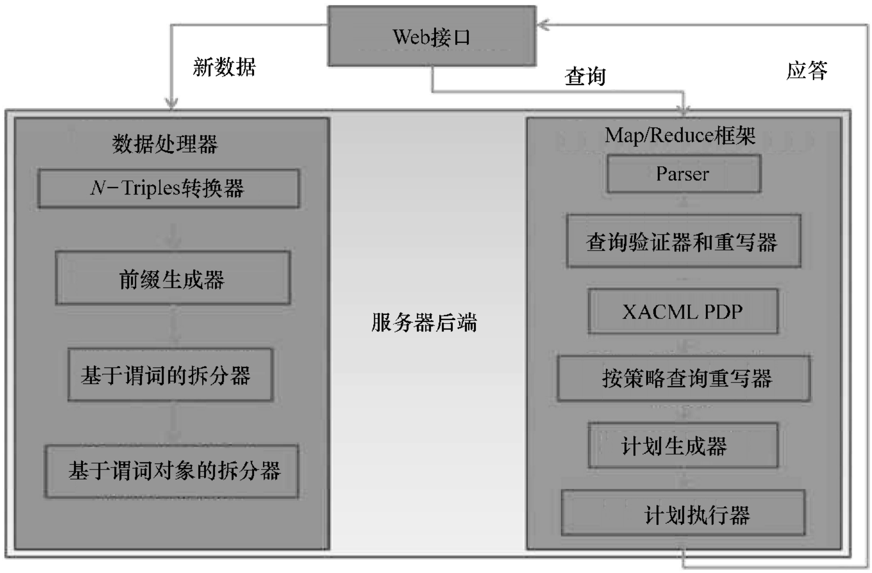


图 13.4 操作架构

13.4.1 数据生成和存储

对于我们的实验，我们使用 LUBM（见本章参考文献［GUO05］）数据集。它是一个基准数据集，旨在使研究者能够评估语义 Web 存储库的性能（见本章参考文献［GUO04］）。LUBM 数据生成器以 RDF/XML 序列化格式生成数据。这种格式不适合实现我们的实验，因为我们将数据存储在 HDFS 中，以平面文件的形式存在，而要检索一个单一的三元组，我们需要解析整个文件。因此，我们将数据转换为 *N* – Triples 以存储数据。因为使用该格式，我们在文件的一行中就会具有完整的 RDF 三元组（主体、谓词和客体），这对于使用其进行 Map/Reduce 作业非常方便。下面几节介绍了将数据转换为预期格式的处理步骤。

13.4.2 文件组织

由于在 Hadoop 和 Map/Reduce 框架中，一个文件是 Map/Reduce 作业的最小输入单位，因此我们并不需要将数据存储在单个文件中，且在没有缓存的情况下，始终从磁盘读取文件。如果我们将所有数据放到一个文件中去，那么每次查询时都要将整个文件输入到作业中去。相反，我们将数据分成多个较小的文件。拆分分为两个步骤，我们将在后面部分有所讨论。

13.4.3 谓词拆分

在第一步中，我们根据谓词划分数据。该划分立即使我们能够减少 SPARQL 查询任何没有变量谓词的搜索空间。对于这样的查询，我们可以为每个谓词选择一个文件，并仅在这些文件上执行查询。为简单起见，我们使用谓词命名文件。例如，包含谓词 p1

的所有三元组：pred 进入名为 p1 – pred 的文件。但是，如果在三元组模式（见本章参考文献［W3e］）中有变量谓词且无法确定客体的类型，则必须考虑所有文件。如果可以确定客体的类型，那么我们考虑具有该客体类型的所有文件。在 13.5 节中我们就此进行了更多的讨论。在实际的 RDF 数据集中，不同谓词的数量通常不是很大（见本章参考文献［STOC08］）。然而，数据集中的谓词数量往往很多。在这种情况下，系统性能不会有所不同，因为我们只是选择与 SPARQL 查询中指定的谓词相关的文件。

13.4.4　使用客体的显式类型信息拆分

在下一步中，我们使用 rdf_type 文件中显式类型的信息。谓词 rdf：type 用于 RDF 表示资源是类的实例。首先将 rdf_type 文件分成与 rdf：type 谓词所具有不同的客体一样多的文件。例如，在本体中，如果类层次结构的叶节点是 c_1、c_2、…、c_n。然后，我们将为每个叶节点创建文件，文件名将类似于 type_c_1、type_c_2、…、type_c_n。需要注意的是，客体值 c_1、c_2、…、c_n不再需要存储在文件中，因为它们可以容易地从文件名中检索。这进一步减少了存储数据所需的空间量。因此，我们为谓词 rdf：type 的每个不同客体值生成这样一个文件。

13.4.5　使用客体的隐式类型信息拆分

我们根据客体的类型划分剩余的谓词文件。并非所有客体都是统一资源标识符（Uniform Resource Identifier，URI），有些是文字。这些文字保留在由谓词命名的文件中，它们不需要进一步处理。这些文件中没有提到 URI 客体的类型信息，但我们可以从 type_ * 文件中检索。URI 客体移动到它们各自的名为 predicate_type 的文件中。例如，如果三元组具有谓词 p 且 URI 客体的类型为 c_i，则主体和客体将显示在文件 p_c_i 中的一行中。要做这个拆分，我们需要加入一个带有 type_ * 文件的谓词文件来检索类型信息。

我们在 13.5 节描述过 Map/Reduce 框架，它有三个子组件。它从用户中获取 SPARQL 查询，并将其传递给输入和计划生成器。该组件通过使用 13.5 节中描述的算法来选择输入文件，决定需要多少 Map/Reduce 作业，并将信息传递给使用 Map/Reduce 框架运行作业的 Join Executer 组件。然后它将查询到的答案从 Hadoop 中转给用户。

13.5　Map/Reduce 框架

13.5.1　简介

Map/Reduce 框架是我们云计算工作的核心。我们将在本书的多个章节中讨论 Map/Reduce，因为它与本章的内容有关。在本节中，我们将讨论如何使用 Map/Reduce 框架组件来响应 SPARQL 查询。

13.5.2 节我们讨论选择输入文件来响应查询的算法。13.5.3 节描述了生成一个 SPARQL 查询的计划中所需的成本估算。它介绍了我们在以下讨论中使用的几个术语。我们还描述了在评估计划成本中应该遵循的理想模型，并介绍我们在实践中使用的基于

启发式的模型。13.5.4 节提出了基于启发式的贪婪算法，以生成使用 13.5.3 节介绍的成本模型的查询计划。在某些情况下，我们会面临在限定条件下来制定计划。在 13.5.5 节中，我们将讨论如何处理这些特殊情况。13.5.6 节描述了如何通过一个示例查询来实现 Hadoop 与 Map/Reduce 作业的连接。

13.5.2 输入文件选择

在确定作业之前，选择需要输入到作业的文件。我们在查询处理的这个步骤中应用了一些查询重写功能，接受用户提交的查询并遍历三元组模式。我们可能会遇到以下情况：

1）在三元组模式中，如果谓词是变量，则选择所有文件作为作业的输入并终止遍历。

2）如果谓词是 rdf：type，并且客体是具体的，我们选择具有该特定类型的类型文件。例如，对于 LUBM 查询 9（Listing1），我们可以选择文件 type_Student 作为输入集的一部分。然而，这产生了一个有趣的情景。在我们的数据集中，实际上没有名为 type_Student 的文件，因为 Student 类不是本体树中的一个叶节点。在这种情况下，我们参考 LUBM 本体论（见本章参考文献［LEHI］）来确定正确的输入文件集。我们添加文件 type_GraduateStudent、type_ UndergraduateStudent，并且将作为 GraduateStudent 的 type_ResearchAssistant、UndergraduateStudent 和 ResearchAssistant 都是子树的叶节点，其根在学生节点。

3）如果谓词是 rdf：type，并且客体是一个变量，那么如果变量的类型由另一个三元组模式定义，我们选择具有该特定类型的类型文件。否则，我们选择所有类型的文件。

4）如果谓词不是 rdf：type，并且客体是一个变量，那么我们需要确定客体的类型是否由查询中的另一个三元组模式指定。在这种情况下，我们可以重写查询来消除一些连接。例如，在 LUBM Query 9（Listing1）中，*Y* 的类型被指定为院系和 *Z* 被指定为课程，这些变量用作最后三个三元组模式中的客体。如果我们选择文件 advisor_Lecturer、advisor_PostDoc、advisor_FullProfessor、advisor_AssociateProfessor、advisor_AssistantProfessor 和 advisor_ VisitingProfessor 作为输入集的一部分，那么第 2 行的三元组模式就变得不必要了。类似地，如果文件采取 Course_Course 并且选择 Course_ GraduateCourse，则第 3 行中的三元组模式变得不必要。因此，我们得到如 Listing2 所示的重写查询。但是，如果没有指定客体的类型，那么我们需要选择该谓词的所有文件。

5）如果谓词不是 rdf：type，并且客体是具体的，那么我们应选择该谓词的所有文件。

```
Listing 1. LUBM Query 9
SELECT ?X ?Y ?Z WHERE {
?X rdf:type ub:Student.
?Y rdf:type ub:Faculty.
?Z rdf:type ub:Course.
?X ub:advisor ?Y.
?Y ub:teacherOf ?Z.
?X ub:takesCourse ?Z}
```

```
Listing 2. Rewritten LUBM Query 9
SELECT ?X ?Y ?Z WHERE {
?X rdf:type ub:Student.
?X ub:advisor ?Y.
?Y ub:teacherOf ?Z.
?X ub:takesCourse ?Z}
```

13.5.3 查询处理的成本评估

我们运行 Hadoop 作业来应答 SPARQL 查询。在本节中，我们将讨论如何评估工作成本。然而，在这之前，需介绍一些以后使用的定义：

定义 13.1

三元组模式，TP：三元组模式是由出现在 SPARQL 查询 WHERE 子句中的主体、谓词和客体构成的有序集合。主体、谓词和客体可以是变量（无界）或具体值（有界）。

定义 13.2

三元组模式连接，TPJ：三元组模式连接是两个可变三元组模式之间的连接。

定义 13.3

Map/Reduce 连接，MRJ：Map/Reduce 连接是一个变量上的两个或更多个三元组模式之间的连接。

定义 13.4

作业，JB：作业 JB 是一个 Hadoop 作业，在这里要做一个或多个 MRJ。JB 有一组输入文件和一组输出文件。

定义 13.5

Conflicting Map/ReduceJoins，CMRJ：冲突的 Map/Reduce 连接是一对共享三元组模式的不同变量上的 MRJ。

定义 13.6

Nonconflicting Map/ReduceJoins，NCMRJ：非冲突的 Map/Reduce 连接是一对 MRJ，不共享任何三元组模式或共享三元组模式，并且 MRJ 位于同一变量上。

用一个例子会更好地说明这些术语。在 Listing3 中，我们显示 LUBM Query 12 第 2、3、4、5 行各有一个三元组模式。第 2 行和第 4 行的 TP 在变量？X 之间的连接是 MRJ。如果我们做两个 MRJ，一个在第 2 行和第 4 行的 TP 之间的变量？X 上，另一个在第 4 行和第 5 行的 TP 之间的变量？Y 上，第 4 行将有一个 CMRJ 作为 TP（？X ub：works-For？Y）参与两个不同变量 *X* 和 *Y* 的两个 MRJ。这种类型的连接称为 CMRJ，因为在 Hadoop 作业中，TP 的多个变量不能同时成为一个键，并且在该键上执行 MRJ。所示的 NCMRJ 将是在变量？X 上的第 2 行和第 4 行中的三元组模式之间的一个 MRJ，而在变量？Y 上的行 3 和 5 中的三个模式之间是另一个 MRJ。这两个 MRJ 可以组成一个 JB。

```
Listing 3. LUBM Query 12
SELECT ?X WHERE {
?X rdf:type ub:Chair.
?Y rdf:type ub:Department.
```

```
?X ub:worksFor ?Y.
?Y ub:subOrganizationOf http://www.U0.edu}
```

13.5.3.1 理想模型

为了响应 SPARQL 查询，我们可能需要多个作业。因此，在理想的情况下，用于处理查询的成本估算需要对每个响应该查询的作业进行单独的成本评估。一个作业包含三个主要任务，即读、排序和写。我们根据这三项任务去评估一份作业的成本。对于每个任务，单位成本被分配给它处理的每个三元组模式。在当前的模型中，我们假设读和写的成本是一样的。

$$\text{Cost} = \left(\sum_{i=1}^{n-1} \text{MI}_i + \text{MO}_i + \text{RI}_i + \text{RO}_i\right) + \text{MI}_n + \text{MO}_n + \text{RI}_n \tag{13.1}$$

$$= \left(\sum_{i=1}^{n-1} \text{Job}_i\right) + \text{MI}_n + \text{MO}_n + \text{RI}_n \tag{13.2}$$

$$\text{Job}_i = \text{MI}_i + \text{MO}_i + \text{RO}_i + \text{RI}_i \quad （如果\ i < n） \tag{13.3}$$

这里，MI_i 是 Job_i 的映射输入阶段，MO_i 是 Job_i 的映射输出阶段，RI_i 是 Job_i 的规约输入阶段，RO_i 是 Job_i 的规约输出阶段。

式（13.1）是处理查询的总成本。它是每个作业的单个成本总和，并且是最终作业的唯一映射阶段。我们不考虑最终作业的规约输出成本，因为当这个输出是查询和给定数据集的最终结果时，任何查询计划都是相同的。作业本质上对文件数据执行 Map/Reduce 任务。式（13.2）表达了将 Map/Reduce 任务划分为子任务。因此，为了估算每个作业的成本，我们将合并每个子任务的估算成本。

映射输入（Map Input，MI）阶段。此阶段从存储在 HDFS 的选定输入文件中读取三元组模式。因此，我们可以用每个选定文件中三元组的总体数量去估计 MI 阶段的开销。

映射输出（Map Output，MO）阶段。MO 阶段的估计取决于正在处理的查询类型。如果查询没有约束变量（例如［？X ub：worksFor？Y］），则映射阶段的输出等于输入。所有三元组模式都转换为键值对，并作为输出。因此，对于这样的查询，MO 成本将与 MI 成本相同。然而，如果查询涉及绑定变量（例如，［？Y ub：subOrganizationOf <http：//www.U0.edu>］），则在创建键值对之前，可以应用组件选择性评估。所得到的三元组模式估算将考虑到 MO 阶段的成本，所选的三元组也将写入本地磁盘。

规约输入（Reduce Input，RI）阶段。在这个阶段，通过 HTTP 读取来自 MO 阶段的三元组，然后根据它们的键值对进行排序。排序后，具有相同键的三元组分组在一起。因此，RI 阶段的成本估计等于 MO 阶段。在 RI 阶段中排序的键值对的数量等于在 MO 阶段中生成的键值对的数量。

规约输出（Reduce Output，RO）阶段。RO 阶段涉及执行链接。因此，在这个阶段，我们可以使用链接三元组模式选择性汇总统计信息，来估计其输出的大小。下面，我们详细讨论我们框架所需的链接三元组的模式选择性汇总统计信息。

然而，在实践中上述讨论仅适用于第一个作业。对于后续作业，我们缺乏精确的知识，也不能在第一个作业的应用加入链接后对所选择的三元组模式数量进行估计。因此，对于这些作业，我们可以将第一个作业中 RO 阶段的大小作为后续作业中不同阶段的上限。

式（13.3）显示了非常重要的假设。它说明了中间作业的总成本，当 $i<n$，在计算总成本时应该包括 RO 阶段的成本。

13.5.3.2　启发式模型

在本节中，我们展示的理想模型是不实际的或是不具有成本效益优势的。这有几个问题使得理想模型在实践中很少被关注。首先，理想模型考虑了简单的抽象成本，即不同阶段读取和写入的三元组数量，忽略了复制、排序等其他情况的实际成本，也忽略了这些三元组以及在 Hadoop 中运行作业的开销。但是将这些成本精确地纳入模型中是一项艰巨的任务。能做出相当好的评估，或许是一项不平凡的工作。第二，为了估计中间加入的输出，我们需要保持全面的汇总统计。在 Hadoop 中的 Map/Reduce 作业中，变量的所有连接都连接在一起。例如，在重写 LUBM 查询 9（Listing 2）中，变量 X 上有三个连接。当运行作业以对 X 进行连接时，X 上的所有连接在三模式 1、2 和 4 之间完成。如果 X 上有三个以上的连接，则所有连接都将在一个作业中处理。这表明要收集汇总统计信息，来估计连接选择性，这时我们将面临指数量级的连接情况。例如，在具有 p1、p2 和 p3 的三元组模式之间，可能存在 23 种类型的连接，因为在每个三元组模式中，变量既可以作为主体也可以作为客体。在重写 Query 9 的情况下，它是 1、2 和 4 之间的主体—主体—主体连接。这三个之间可以有更多类型的连接，例如主体—客体—主体、客体—主体—客体等。这意味着，在 P 谓词之间，单个变量上可以有 2^P 种类型的连接（忽略变量可能同时出现为三元组模式中主体和客体的情况）。如果数据集中有 P 个谓词，那么就需要收集汇总统计，全部的连接数量可以用如下公式计算

$$2^2 \times C_2^P + 2^3 \times C_3^P + \cdots + 2^P \times C_P^P$$

在 LUBM 数据集中，有 17 个谓词。总共有 129140128 连接情况，它是一个庞大的数据。收集如此大量连接情况的汇总统计数据将是非常消耗时间和空间的。因此，我们采取了另一种方法。

我们注意到，在 Hadoop 中运行作业有很大的开销。因此，如果我们最大限度地减少应答查询的作业，那么就可以获得最快的方案。这些开销是由几个磁盘 I/O 和网络传输引起的，他们可能是任何 Hadoop 作业的组成部分。当作业提交到 Hadoop 集群时，至少要执行以下一组操作：

1）可执行文件从客户端传输到 Hadoop JobTracker（见本章参考文献［WIKIa］）。

2）JobTracker 决定 哪一个 TaskTrackers（见本章参考文献［WIKIb］）将执行这项工作。

3）可执行文件通过网络分发给 TaskTrackers。

4）从 HDFS 读取数据就是映射过程的开始。

5）将映射输出写入磁盘。

6）从磁盘读取映射输出，无序传输（运行规约进程，通过网络传输到 TaskTrackers）、排序，并写入磁盘。

7）通过读取磁盘的输入来启动规约进程。

8）将规约输出写入磁盘。

即使对于少量数据，这些磁盘操作和网络传输也是昂贵的。例如，在我们的实验中，可以观察到，一个作业所带来的额外开销几乎等同于读取十亿个的三元组。原因是在每个作业中，映射进程的输出总是在规约进程之前进行排序。即使用户不需要这种排

序也是不可避免的。因此，在 n 个作业中处理成百上千万三元组的成本将会更低，而不是在 $n+1$ 作业中处理数个较少的三元组。

为了进一步研究，我们做了一个实验，使用了 Listing4 所示的查询。在？Z 上的 TP 2 和 3 之间的连接选择性是最高的。因此，一个查询计划生成的算法为第一个作业选择一个连接时，需要使用选择性因子选取连接。由于其他 TP 1 和 4 与 TP 2 或 3 共享变量，因此它们不能参加任何其他的连接；而且，它们不共享任何变量，因此可以在此作业中执行的唯一可能的连接是？X 上的 TP2 和 3 之间的连接。一旦这个连接完成，剩余两个连接就是：一个在 TP 1 和关于变量？X 的第一个作业的连接输出之间，一个在 TP 4 和关于变量？Y 的第一个作业的连接输出之间。我们发现第一个连接的选择性大于后者。因此，第二个作业将会加入，TP 4 将再次不参加。在第三个和最后一个作业中，第二个作业的连接输出将与关于变量？Y 上的 TP 4 相连接。这是使用连接选择性评估生成的方案。但最小作业方案是一个两作业计划，其中第一作业连接到关于？X 的 TP 1 和 2 以及关于？Y 上的 TP 3 和 4。第二个和最后的作业连接到关于？Z 的第一个作业的两个连接输出。我们得到的查询运行时间在本章参考文献［HUSA11a］里给出。

```
Listing 4. Experiment Query
?S1 ub:advisor ?X.
?X ub:headOf ?Z.
?Z ub:subOrganizationOf ?Y.
?S2 ub:mastersDegreeFrom ?Y
```

对于每个数据集，即使三作业方案由于连接选择性顺序而产生较少的中间数据，我们也能发现两作业计划比三作业计划更快。我们也可以通过另一个小型实验中的观察结果，来解释这一点。我们生成了包含随机整数，大小为 5MB 和 10MB 的文件。将文件放在 HDFS 中。对于每个文件，我们首先通过程序读取文件并记录所需的时间。在读取时，我们的程序读取文件的三个可用副本中的一个。然后，我们运行了一个 Map/Reduce 作业，重新编写已排序数字的文件。然后，使用 Map/Reduce 排序来排序输出。请注意，当它写入文件时，会写入三个重复的文件。我们发现对于 5 MB 文件，Map/Reduce 作业读取、排序和写入的时间要长达 24.47 次。对于 10 MB 的文件，它是 42.79 次。这清楚地显示了 Map/Reduce 作业的写入和数据传输操作比仅从一个副本的简单读取更昂贵。由于作业的数量，三作业方案需要进行更多的磁盘读写操作以及网络数据传输，因此即使读取较少的输入数据，其结果也比两作业方案慢。

由于这些原因，我们不追求理想的模式。我们遵循实用模式，即生成具有最少可能作业的查询方案。然而，在生成最小作业方案的同时，每当我们需要在一个作业中的多个连接中的选择要考虑的连接，而不是随机选择时，我们都要使用汇总连接统计信息。这在 13.5.4 节中有描述。本章参考文献［HUSA11a］中提供了实验结果的有关图表的更多细节。

13.5.4 查询计划生成

在本节中，我们首先定义查询方案生成问题，并说明为理想模型以及实际模型生成最佳（即最低成本）查询的方案在计算上是昂贵的。然后，我们将提出一个启发式和一个贪婪方法来产生一个近似的解决方案来产生最佳方案。

运行示例

我们将使用以下查询作为本节中的一个运行示例：

```
Listing 5. Running Example
SELECT ?V,?X,?Y,?Z WHERE{
?X rdf:type ub:GraduateStudent
?Y rdf:type ub:University
?Z ?V ub:Department
?X ub:memberOf ?Z
?X ub:undergraduateDegreeFrom ?Y}
```

为了简化表示，我们只会用该模式的变量来引用TP。例如，第一个TP（？X rdf：type ub：GraduateStudent）将被简单表示为X。此外，在简化版本中，整个查询将表示如下：{X、Y、Z、XZ、XY}。我们将使用符号连接（XY，X）表示公共变量X上的两个TP（XY和X）之间的连接操作。

定义13.7

（最低成本方案生成问题）。（最佳方案问题又名Bestplan问题）。对于给定的查询，最佳方案问题是生成一个作业方案，以便最大限度地减少作业的总成本。请注意，最佳方案考虑了更一般的情况，其中每个作业都有与其相关的成本（即理想模型）。

例如在我们的运行示例中，有两个可能的作业方案如下所示：

方案1. job_1 =，{X、XY、XZ}

则有TP =，{YZ，YZ} . job_2 =，{Y，YZ}，

则有TP =，{Z，Z} . job_3 =，{Z，Z}。总成本 =，成本（job_1）+ 成本（job_2）。

方案2. job_1 =，{XZ、Z} 和连接（XY、Y）

则有TP =，{X、X、X} . job_2 =，连接（X、X、X）。

总成本 =，成本（job_1）+ 成本（job_2）。

最佳方案问题是在所有可能的作业方案中找到最低成本的作业方案。接下来，我们定义一些相关的术语，并讨论方法的复杂性。

定义13.8

（连接变量）。两个或多个三元组模式中常见的变量。例如，在前面的运行示例查询中，X、Y、Z是连接变量，但V不是。

定义13.9

（完全消除）。一个消除连接变量的连接操作。例如，在示例查询中，如果我们加入（XY，Y），则可以完全消除Y。

定义13.10

（部分消除）。一个部分消除连接变量的连接操作。例如，在示例查询中，如果我们在相同的作业中执行连接（XY，Y）和连接（X，ZX），则所得到的三元组模式将是{X，Z，X}。因此，Y将被完全消除，但X将被部分消除。所以，连接（X，ZX）执行部分消除。

定义13.11

E－Count（V）。E－count（V）是在完全消除变量V之后得到的三元组模式中的连接变量的数量。在运行示例中，连接（X，XY，XZ）完全消除了X，并且得到的三元组

模式（*YZ*）具有两个连接变量 *Y* 和 *Z*，所以 E－count（*X*）＝2. 同样，E－count（*Y*）＝1，E－count（*Z*）＝1。

13.5.4.1 最佳方案的计算复杂度

可以看出，生成最低成本查询方案在计算上是代价很高的，因为搜索空间是指数级的。首先，我们将这个问题公式化，然后给出其复杂性。

13.5.4.1.1 问题公式化

我们把 Bestplan 作为搜索问题进行公式化。令 $G=(V, E)$ 为加权有向图，其中每个顶点 $v_i \in V$ 表示三元组模式的状态，每个边 $e_i \in (v_{i1}, v_{i2}) \in E$ 表示从 statev_{i1} 到 statev_{i2} 的转换。v_0 是没有执行连接的初始状态，即给定的查询。此外，v_{goal} 是目标状态，它代表了所有连接执行的三元组模式的状态。问题是找到从 v_0 到 v_{goal} 的最短加权路径。

例如，在我们的运行示例查询中，初始状态 $v_0=$，$\{X, Y, Z, XY, XZ\}$，目标状态为 $v_{goal}=\varnothing$，即不再有三元组模式。假设第一个作业（job_1）执行连接（*X*，*XY*，*XZ*）。然后，结果三元组模式（新状态）将是 $v_1=$，$\{Y, Z, YZ\}$，并且 job_1 通过 edge（v_0，v_2）表示。edge（v_0，v_1）的权重是 $job_1=\text{cost}(job_1)$ 的成本，其中 cost 是给定的成本函数。图 13.4 显示了示例查询的部分图。

13.5.4.1.2 搜索空间大小

给定图 $G=(V,E)$，Dijkstra 的最短路径算法可以在 $O(|V|\log|V|+|E|)$ 时间内找到从源到所有其他节点的最短路径。然而，对于 Bestplan，可以看出，在最坏的情况下，$V \geqslant 2^K$，其中 *K* 是给定查询中连接变量的总数。因此，图中顶点的数量是指数级的，导致指数级搜索问题。在本章参考文献［HUSA11a］中，我们已经表明，Bestplan 问题的最坏情况的复杂性是 *K* 指数级的，即给定查询中加入变量的数量。

13.5.4.2 条件放宽的最佳方案问题和近似解决方案

在条件放宽的最佳方案问题中，我们假定所有作业有统一的成本。虽然这种条件放宽并不会减少搜索空间大小，但问题在于找到具有最少作业数量的作业方案。请注意，这是模型应用到实际的问题。

定义 13.12

（条件放宽的最佳方案问题）。条件放宽的最佳方案问题是找到包含最少作业的作业方案。

接下来，我们说明如果合理选择连接，并且不满足条件的连接操作在作业中被撤销，那么我们可以为任何查询设定所需的最大作业数量的上限。然而，生成所有可能的作业方案仍然在计算上是昂贵的。因此，我们采用一种贪婪算法（算法 1），找出“条件放宽的最佳方案问题”的近似解决方案，保证在上限范围内找到作业方案。

定义 13.13

（尽早消除启发式）。尽早消除启发式尝试在每项作业中尽可能多地做到完全消除。

这个启发式留下了最少数量的变量来连接到下一个作业。为了应用这个启发式算法，我们首先必须在 E－count 最少的每个作业中选择变量。算法 1 中应用了这个启发式算法。

算法 1：条件放宽的最佳方案（查询）

1：$Q \leftarrow$ Remove non－joining variables（Q）

2：while $Q \neq$ Empty do

```
3: J←1//作业总数
4: U = , {u_1, …, u_K} ←所有变量均按它们的 E-counts 值进行非降序排列
5: Job_J←Empty//当前作业的连接操作列表
6: tmp←Empty //临时存储中间结果
   //三元组模式
7: for i = 1to K do
8: if Can-Eliminate (Q, u_i) = true then
   //尽可能地完全或者部分消除
9: tmp←tmp ∪ Join-result (TP (Q, u_i))
10: Q←Q-TP (Q, u_i)
11: Job_J←Job_J ∪ join (TP (Q, u_i))
12: end if
13: end for
14: Q←Q ∪ tmp
15: J←J+1
16: end while
17: return {Job_1, …, Job_{J-1}}
```

算法 1 的描述：该算法从查询 Q 中删除所有非连接变量开始。在运行示例中，$Q=\{X, Y, VZ, XY, XZ\}$，并移除非连接变量 V 使 $Q=\{X, Y, Z, XY, XZ\}$。在 while 循环中，从 Job_1 开始生成作业方案。在第 4 行中，我们根据它们的 E-count 对变量进行排序。排序变量为 $U=\{Y, Z, X\}$，因为 Y 和 Z 有 E-count = 1，X 有 E-count = 2。对于每个作业，连接操作列表存储在变量 Job_J 中，其中 J 是当前作业的 ID。另外，使用临时变量 tmp 来存储要在当前作业中执行的连接的结果三元组（行 6）。在 for 循环中，检查每个变量以查看变量是否可以完全或部分消除（第 8 行）。如果是，我们将连接结果存储在临时变量（第 9 行）中，更新 Q（第 10 行）将此连接添加到当前作业（第 11 行）。在我们运行的例子中，会有以下操作：for 循环的迭代 1：$u_1=(Y)$ 可以彻底消除。这里，$TP(Q, Y)$，包含迭代的 Q 中的三元组模式的 for 循环：$u_3=(X)$ 不能完全或部分消除，因为没有剩余的 TP 可以加入。因此，当 for 循环终止时，我们有 $job_1=\{join(Y, XY), join(Z, XZ)\}$ 和 $Q=\{X, X, X\}$。在 while 循环的第二次迭代中，我们将有 $job_2=\{X, X, X\}$。因为在这个连接之后，Q 变成空，所以退出 while 循环。最后，从算法返回 $\{job_1, job_2\}$。

在本章参考文献［HUSA11a］中，我们已经证明，对于任何给定的查询 Q，包含 K 个连接变量和 N 个三元组模式，条件放宽的最佳方案算法（Q）生成包含最多 J 个作业的作业计划，其中

$$J=\begin{cases} 0 & N=0 \\ 1 & N=1 \text{ 或 } K=1 \\ \min(\lceil 1.71\log_2 N \rceil, K) & N, K>1 \end{cases} \tag{13.4}$$

13.5.5 通过汇总统计分析突破限制

我们经常遇到需要为多个连接选项选择连接的情况。当两个查询方案（即，连接排

序）都需要最少数量的作业时，可能会发生这些选择。例如，Listing6 所示的查询提出了这种情况。

```
Listing 6. Query Having Tie Situation
?X rdf:type ub:FullProfessor.
?X ub:advisorOf ?Y.
?Y rdf:type ub:ResearchAssistant.
```

查询中的第二个三元组模式使得我们不可能仅使用一个作业回答和解决查询。只有两个可能的方案：可以先将 X 上的前两个三元组模式进行连接，然后将其输出与 Y 上的最后一个三元组模式相连接；或者，我们可以先将 Y 上最后两个模式进行连接，然后将其输出与 X 上的第一个模式相连接。在这种情况下，我们使用一对谓词的连接汇总统计信息而不是随机选择第一个作业的连接变量。我们为第一个作业选择连接，这更有选择性地突破限制。我们使用的连接汇总统计信息在本章参考文献［STOC08］中有描述。

13.5.6　Map/Reduce 连接执行

在本节中，我们将讨论如何使用 Hadoop 的 Map/Reduce 框架来实现回应 SPARQL 查询所需的连接。算法 1 决定了回应查询所需的作业数。它返回一组有序的作业集。每个作业都有相关的输入信息。Map/Reduce 框架的作业处理器组件按照它们在有序集中显示的顺序运行作业。一个作业的输出文件是下一个作业的输入文件。最后一个作业的输出文件带有回应查询的答案。

```
Listing 7. LUBM Query 2
SELECT ?X, ?Y, ?Z WHERE {
?X rdf:type ub:GraduateStudent.
?Y rdf:type ub:University.
?Z rdf:type ub:Department.
?X ub:memberOf ?Z.
?Z ub:subOrganizationOf ?Y.
?X ub:undergraduateDegreeFrom ?Y}
```

Listing 7 显示了 LUBM Query 2，我们将使用它来说明使用 map 和 reduce 方法进行连接的方式。这个查询在变量 X、Y 和 Z 上有 6 个三元组模式和 9 个连接。

我们的输入选择算法选择文件 type_GraduateStudent、type_University、type_Department、具有前缀 memberOf 的所有文件、具有前缀 subOrganizationOf 的所有文件和具有前缀 under GraduateDegree From 的所有文件作为输入，输入到需要回答查询的作业中。

查询方案有两个作业。在 job_1 中，第 2、5 和 7 行的三个模式连接在 X 上，第 3 和 6 行的三元组模式连接在 Y 上。在 job_2 中，第 4 行的三元组模式与关于 Z 的前两个连接的输出相连接，并且 job_1 的连接输出要连接到 Y 上。

job_1 的输入文件是 type_GraduateStudent、type_University、具有前缀 memberOf 的文件、具有前缀 subOrganizationOf 的文件，以及所有带有前缀的文件他们都在 under GraduateDegreeFrom。在映射阶段，我们首先对输入值进行标记，实际上是输入文件的一行。然后，我们检查输入文件名，如果输入是来自 type_GraduateStudent，我们输出一个键值对，其主体 URI 前缀为 X#，键和标识字符串 GS#作为值。该值用作标记，表示该键的

类型为 GraduateStudent。主体 URI 是令牌分发器返回的第一个令牌。类似地，对于来自文件 type_University 的输入，输出具有以 Y#为前缀的主体 URI 的键值对，键和标志串 U#作为其值。如果来自任何文件的输入具有前缀 memberOf，则我们通过令牌分发器从输入行中检索主体和客体，并输出一个键值对，其主体 URI 以 X#为前缀，键和以 MO#为前缀的客体作为其值。对于具有前缀 subOrganizationOf 的文件的输入，我们输出键值对，使客体以 Y#为前缀，键和以 SO#为前缀的主体作为其值。对于具有 under Graduate DegreeFrom 前缀的文件的输入，我们输出键值对，使主体 URI 以 X#为前缀，键和以 UDF#为前缀客体作为其值。因此，基于正在进行的连接，为主体或客体做好了映射输出键。这就是为什么对于来自带有前缀 subOrganizationOf 的文件的三元组，用客体作为键的原因。显然，连接型变量 Y 是第 6 行中三元组模式中的客体。对于所有其他输入，由于连接变量 X 和 Y 是第 2、3、5 和 7 行中三元组模式中的主体，主体就被指定为键。

在“规约阶段”中，Hadoop 对单个键的所有值进行分组，并为每个键提供关键字，为值的集合提供迭代器。通过考察前缀，我们可以立即知道它是否是 X 或 Y 的值。在任何一种情况下，我们使用相同的键输出一个键值对，并将所有值连接起来，使其成为一个字符串值。因此，在这个规约阶段之后，X 上的连接完成，Y 的连接部分完成。

job_2 的输入文件是 type_Department 文件和 job_1 输出文件（也就是 job_1 . out 文件）。就像 job_1 的映射阶段，在 job_2 的映射阶段，我们也标记输入值，这实际上是输入文件的一行。然后，我们检查输入文件名，如果输入来自 type_Department，则我们将输出一个键值对，其主体 URI 以 Z#前缀，键和标志字符串 D # 作为值。如果输入来自 job_1. out，我们发现有前缀 Z# 的值。我们将该值作为输出键，并将其余值串连成一个字符串，并使其成为输出值。基本上，我们将 Z# 值作为键连接到 Z 上。

在规约阶段，我们知道键就是 Z 的值。值集合有两种类型的字符串。一个有 X 值，它们是研究生的 URI，以及他们获得本科学位的 Y 值。Z 值，即关键字，可能是也可能不是 Y 值的子组织。其他类型的字符串只有 Y 值是大学，Z 值是子组织。我们对值集合进行迭代，然后在 Y 值上加入两种类型的元组。从连接输出，我们找到具有 X、Y 和 Z 值的结果元组。

13.6 结果

在介绍我们的评估结果之前，首先介绍我们实验的基准数据集，然后是为了比较而进行的备选存储库评估，以及详细的实验设置。

13.6.1 数据集、框架和实验配置

13.6.1.1 数据集

在我们使用 SPARQL 查询处理的实验中，使用两个合成数据集：LUBM（见本章参考文献［GUO05］）和 SP2B（见本章参考文献［SCHM09］）。LUBM 数据集通过使用本体论（见本章参考文献［LEHI］）生成关于大学的数据。它有 14 个标准查询。一些查询需要推理获得答案。LUBM 数据集非常适用于推理和可扩展性测试。对于所有 LUBM 数据集，我们使用默认的基础数据。SP2B 数据集适用于具有复杂查询和数据访问模式的可扩展性测

试。它有 16 个查询，其中大多数具有复杂的结构。

13.6.1.2 基准框架

将我们的框架与 RDF－3X（见本章参考文献［NEUM08］）、Jena（见本章参考文献［JENA］）和 BigOWLIM（见本章参考文献［ONTO］）进行比较如下。RDF－3X 被认为是具有永久存储的最快的语义 Web 框架。Jena 是语义 Web 数据的开源框架。它具有可用于存储和检索 RDF 数据的一些模型。我们选择了 Jena 的内存和 SDB 模型来比较我们的框架。顾名思义，内存模型将数据存储在主存储器中，并且不保留数据。SDB 模型是一种持久的模型，可以使用许多现成的数据库管理系统。我们在实验中使用 MySQL 数据库作为 SDB 的后端。BigOWLIM 是一个专有的框架，是语义 Web 数据的最先进的显著快速框架，它可以作为持久的和非分散的存储。所有这些框架都运行在单机配置中。

13.6.1.3 实验配置

硬件：我们有一个用于框架的 10 节点的 Hadoop 集群。每个节点均具有以下配置：Pentium IV 2.80GHz 处理器、4GB 主内存和 640GB 磁盘空间。我们在具有 2.80GHz 四核处理器、8GB 主内存和 1 TB 磁盘空间的强大单机上运行 Jena、RDF－3X 和 BigOWLIM 框架。

软件：我们使用 Hadoop－0.20.1 作为框架。将我们的框架与 Jena－2.5.7 进行了比较，Jena－2.5.7 使用 MySQL 14.12 作为其 SDB 模型，我们使用 BigOWLIM 版本 3.2.6。对于 RDF－3X，我们使用版本 0.3.5 的源代码。

13.6.2 评估

针对我们目前的框架、RDF－3X、Jena 内存和 SDB 模型以及 BigOWLIM，我们进行了它们之间的性能比较，更多细节见本章参考文献［HUSA11a］。我们使用三个 LUBM 数据集：10000、20000 和 30000，分别有 11 亿、22 亿和 33 亿个三元组。对于 RDF－3X，为了加载数据，初始种群时间分别为 655、1756 和 3353 分钟。这表明 RDF－3X 加载时间呈指数增长。LUBM（30000）有三倍于 LUBM（10000）的三元组，加载时间则需要五倍以上。

为了进行评估，在本工作中我们选择了 LUBM 查询 1、2、4、9、12 和 13 进行汇报。这些查询提供了一个很好的混合，包括简单和复杂的结构，推理和多种类型的连接。他们是基准测试中其他查询的代表，因此这里仅报告这些涵盖了遗漏的查询中含有的所有类型的变体，也节省了空间。查询 1 是一个简单的选择性查询。针对这个查询，RDF－3X 比 HadoopRDF 快得多。RDF－3X 使用 6 个指标（见本章参考文献［NEUM08］），这 6 个指标实际上构成了数据集。索引为 RDF－3X 提供了一种非常快速的方法来查找三元组，类似于哈希表。因此，RDF－3X 有效地回应了高度选择性的查询。查询 2 是具有复杂结构、低选择性和无约束客体的查询，结果集相当大。对于该查询，对于所有三种数据集的大小，HadoopRDF 的性能优于 RDF－3X。当数据集大小为 33 亿三元组时，RDF－3X 根本无法回应查询。RDF－3X 返回内存分段错误消息，不产生任何查询结果。查询 4 也是一种高度选择性的查询，也就是说，由于第二个三元组模式中的绑定客体，结果集很小，但是它需要通过推论来回应它。第一个三元组模式使用类 Person，它是许多类的超类。LUBM 数据集中的资源不是 Person 类型，而是它的子类型，有很多资源。RDF－3X 不支持推理，所以我们不得不将查询转换为具有一些联合操

作的等效查询。对于此查询，RDF－3X 胜过 HadoopRDF。查询 9 在结构上与查询 2 类似，但是它需要大量的推理进行判断。此查询的前三个三元组模式使用在数据集中未明确实例化的类。然而，数据集中包括相应子类的许多实例。它也是需要最大数据集连接的查询，而且能从我们评估的查询中返回最大的结果集。对于 11 亿三元组的数据集，RDF－3X 的速度比 HadoopRDF 快，但是对于其他两个数据集，它完全不能回应查询。查询 12 类似于查询 4，因为它是选择性的并且以一个三元组模式进行推理。针对此查询，RDF－3X 完败 HadoopRDF。查询 13 只有两个三元组模式。它们都涉及推理。第二个三元组模式中有一个限定的主体。它返回第二大结果集。对所有数据集的该查询，HadoopRDF 都胜过 RDF－3X。RDF－3X 的性能很慢，因为第一个三元组模式具有非常低的选择性，并且需要较低的选择性连接通过反向链来执行推理。

这些结果使我们得出了一些简单的结论。RDF－3X 对具有高选择性和绑定客体的查询，具有最佳性能。然而，对于具有未绑定客体的查询、低选择性或大型数据集连接，HadoopRDF 的性能优于 RDF－3X。RDF－3X 无法对 33 亿个三元组数据集执行具有未绑定对象的两个查询（查询 2 和 9）。这表明 HadoopRDF 比 RDF－3X 更具可扩展性，可以更有效地处理低选择性查询。

我们还将我们的成果与 Jena 内存模型、SDB 模型以及 BigOWLIM 进行了比较。由于空间和时间限制，我们仅对 LUBM 数据集中的 LUBM 查询 2 和 9 执行了这些测试。我们选择了这些查询，因为它们具有复杂的结构并需要推断。需要注意的是，BigOWLIM 需要 7 GB 的 Java 堆空间来成功地加载 10 亿个三元组数据集。这里只对最大的三个数据集运行 BigOWLIM，因为我们对大数据集的性能更感兴趣。对于每一组，我们分别获得了 Jena 内存模型、Jena SDB 模型、Hadoop 实现和 BigOWLIM 的结果。有时，查询无法完成或内存不足。在大多数情况下，我们的做法是最快的。对于查询 2，Jena 内存模型和 Jena SDB 模型比我们的方法更快，结果分别为 3.9s 和 0.4s。然而，随着数据集越来越大，Jena 内存模型的内存空间已经不足。我们的实施方法要比 Jena SDB 大型数据集模型快得多。例如，对于 1.1 亿的三元组数据，我们的方法花费了 143.5s，而 Jena SDB 模型则为约 5000s。我们发现，Jena SDB 模型无法完成对查询 9 的回应。Jena 内存模型在小数据集中运行良好，但随着数据集的增长，反而变得比我们的方法更慢，并最终耗尽内存。

对于查询 2，在 1.1 亿和 5.5 亿的数据集上，BigOWLIM 比我们的慢。对于 5.5 亿个数据集，它花费了 22693.4s，与其他时间相比是非常高的。对于 10 亿个三元组数据集，BigOWLIM 更快。应该注意的是，我们的框架没有任何索引或三元组缓存，而 BigOWLIM 利用索引，它在启动时加载到主内存中。另外，它也可以将三元组预取到主存储器中。对于查询 9，我们的实现在所有实验中都比 BigOWLIM 快。

应该注意，RDF－3X 查询和 HadoopRDF 查询使用冷运行进行测试。我们这样做的目的是在执行前清除主内存和文件系统缓存。但是，对于 BigWLIM，我们被迫执行热运行。这是因为将数据库加载到 BigWLIM 需要大量时间。因此，我们将一直很容易地在冷运行中胜过 BigOWLIM。所以，我们实际测试了 BigOWLIM 的热运行，而针对 HadoopRDF 进行冷运行。这给 BigOWLIM 带来了巨大的优势，对于大型数据集，HadoopRDF 仍然产生了更好的结果。这表明 HadoopRDF 比 BigOWLIM 更具可扩展性，并且可以为大型数据集提供更有效的查询。

我们进行的最终测试是深入的可扩展性测试。为此，我们对 8 个不同的数据集大小进行了重复的相同查询，一直达到 66 亿。

在我们的实验中，发现查询 1 很简单，只需要一个连接；因此，在所有查询中花费的时间最少。查询 2 是具有最大数量的三元组模式的两个查询之一。由于我们的存储架构，即使它有三倍多的三元组模式，但是也没有占用三倍于查询 1 的应答时间。查询 4 比查询 2 少一个三元组模式，但它需要推理。当我们确定推理的关系时，需要推理的查询在我们的框架中需要更长的时间。查询 9 和 12 也要求推理。细节在本章参考文献［HUSA11a］中给出。

随着数据集规模的增长，回应查询的时间的增加不会成比例增长。时间的增加总是会少一些。例如，10000 个大学是 1000 个大学的数据集三元组的 10 倍，但对于查询 1，时间只增加了 3. 76 倍，查询次数增加了 7. 49 倍。后者是时间上的最大增长，但仍然小于数据集大小的增加。由于空间限制，我们不在这里报告 PS 模式的查询运行时间。实际上，我们发现 PS 模式比 POS 架构慢得多。

13. 7　总结和展望

我们提出了一个能够处理大量 RDF 数据的框架。既然我们的框架是基于 Hadoop 的，就意味着它是一个分布式和高度容错的系统，它自动继承了这两个属性。该框架具有高度可扩展性。为了增加系统的容量，需要做的所有工作就是向 Hadoop 集群添加新的节点。我们还提出了一种存储 RDF 数据的模式，一种限定最差情况下的确定查询处理方案的算法，为了响应 SPARQL 查询，算法中使用了简化的成本模型。实验表明我们的系统具有高度可扩展性。如果我们增加数据量，则引入回应查询的延迟不会按比例增加。结果说明，对于非常大的数据集（超过 10 亿个三元组），如果查询包括低选择性连接或重要推断，HadoopRDF 是更优选和更有效的。如果查询包括产生高选择性的约束客体，其他解决方案可能会更有效。

未来，我们将在多个方面拓展工作。首先，我们将研究更复杂的查询模型。我们将缓存最常见查询的统计信息，并使用动态规划来利用统计信息。其次，我们将评估用户指定的 Hadoop 作业唯一参数的规约器数量对查询运行时间的影响。第三，我们将研究索引条件和二进制格式的进一步使用。最后，我们将处理更复杂的 SPARQL 模式，例如，具有可选块的查询。

参 考 文 献

[ABAD07] Abadi, D.J., A. Marcus, S.R. Madden, and K. Hollenbach, Scalable semantic web data management using vertical partitioning, *Proceedings of the 33rd International Conference on Very Large Data Bases,* 2007, Vienna, Austria.

[ABAD09a] Abadi, D.J. Data management in the cloud: Limitations and opportunities, *IEEE Data Engineering Bulletin*, 32(1): 3–12, March 2009.

[ABAD09b] Abadi, D.J., A. Marcus, S.R. Madden, and K. Hollenbach, SW-store: A vertically partitioned DBMS for semantic web data management, *VLDB Journal*, 18(2): 385–406, April 2009.

[ABOU09] Abouzeid, A., K. Bajda-Pawlikowski, D.J. Abadi, A. Silberschatz, and A. Rasin. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads, *Proceedings of VLDB Endowment*, 2, 922–933, 2009.

[ATRE08] Atre, M., J. Srinivasan, and J.A. Hendler, BitMat: A main-memory bit matrix of RDF triples for conjunctive triple pattern queries, *Proceedings of the International Semantic Web Conference*, 2008, Karlsruhe, Germany.

[BONC06] Boncz, P., T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner, MonetDB/XQuery: A fast XQuery processor powered by a relational engine, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 479–490, 2006, Chicago, IL.

[CARR04] Carroll, J.J., I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, Jena: Implementing the semantic web recommendations, *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters*, pp. 74–83, 2004, New York, NY.

[CHAL] Semantic Web Challenge http://challenge.semanticweb.org.

[CHAN06] Chang, F., J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R.E. Gruber, Bigtable: A distributed storage system for structured data, *Proceedings of the Seventh USENIX Symposium on Operating System Design and Implementation*, November 2006, 205–218, Seattle, WA.

[CHEB07] Chebotko, A., S. Lu, and F. Fotouhi, *Semantics Preserving SPARQL-to-SQL Translation*, Technical Report TR-DB-112007-CLF, 2007.

[CHON05] Chong, E.I., S. Das, G. Eadon, and J. Srinivasan, An efficient SQL-based RDF querying scheme, *Proceedings of the International Conference on Very Large Data Bases (VLDB '05)*, 2005, 1216–1227, Trondheim, Norway.

[CHU07] Chu, C.T., S.K. Kim, Y.A. Lin, Y. Yu, G. Bradski, A.Y. Ng, and K. Olukotun, MapReduce for machine learning on multicore, *Proceedings of Neural Information Processing Systems (NIPS)*, 2006, Vancouver, BC, Canada.

[CLOU] Cloudera University: http://www.cloudera.com/blog/2010/03/how-raytheon-researchers-are-using-hadoop-to-build-a-scalable-distributed-triple-store.

[COMP] National University of Singapore School of Computing: http://www.comp.nus.edu.sg/~epic/.

[CYGA05] Cyganiak, R. *A Relational Algebra for SPARQL*, Technical Report HPL-2005-170, 2005.

[DEAN04] Dean, J. and S. Ghemawat, MapReduce: Simplified data processing on large clusters, *Proceedings of the Sixth Conference Symposium on Operating Systems Design and Implementation*, 2004, 137–150, San Francisco, CA.

[DING05] Ding, L., T. Finin, Y. Peng, P.P. da Silva, and D.L. Mcguinness, Tracking RDF graph provenance using RDF molecules, *Proceedings of the Fourth International Semantic Web Conference*, 2005, Galway, Ireland.

[ELMA94] Elmasri, R. and B. Navathe, *Fundamentals of Database Systems*. Pearson Education, 1994., Boston, MA.

[GUO04] Guo, Y., Z. Pan, and J. Heflin, An evaluation of knowledge base systems for large OWL datasets, *Proceedings of the International Semantic Web Conference*, 2004, Hiroshima, Japan.

[GUO05] Guo, Y., Z. Pan, and J. Heflin, LUBM: A benchmark for OWL knowledge base systems, *Web Semantics: Science, Services and Agents on the World Wide Web*, 3, 158–182, 2005.

[GUO06] Guo, Y. and J. Heflin, A scalable approach for partitioning OWL knowledge bases, *Proceedings of the Second International Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006, Athens, GA.

[HADOa] Apache Software Foundation: http://hadoop.apache.org.
[HADOb] Apache Software Foundation: http://hadoop.apache.org/core/docs/r0.18.3/hdfs_design.html.
[HUSA09] Husain, M.F., P. Doshi, L. Khan, and B. Thuraisingham, Storage and retrieval of large RDF graph using Hadoop and MapReduce, *Proceedings of the First International Conference on Cloud Computing*, http://www.utdal- las.edu/mfh062000/techreport1.pdf, 2009, Beijing, China.
[HUSA10] Husain, M.F., L. Khan, M. Kantarcioglu, and B. Thuraisingham, Data intensive query processing for large RDF graphs using cloud computing tools, *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 1–10, July 2010, Miami, FL.
[HUSA11a] Husain, M.F., J.P. McGlothlin, M.M. Masud, L.R. Khan, and B.M. Thuraisingham, Heuristics-based query processing for large RDF graphs using cloud computing, *IEEE Transactions on Knowledge and Data Engineering*, 23(9): 1312–1327, 2011.
[HUSA11b] Husain, M.F., *Data Intensive Query Processing for Semantic Web Data Using Hadoop and MapReduce*, PhD Thesis, University of Texas at Dallas, May 2011.
[ITEE] The University of Queensland Australia, School of Information Technology and Electrical Engineering: http://www.itee.uq.edu.au/eresearch/projects/biomanta.
[JENA] Apache Software Foundation: http://jena.sourceforge.net.
[KIRY05] Kiryakov, A., D. Ognyanov, and D. Manov, OWLIM: A pragmatic semantic repository for OWL, *Proceedings of the International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS)*, 2005, New York, NY.
[LEHI] Lehigh University: http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl.
[MCGL09] McGlothlin, J.P. and L.R. Khan, RDFKB: Efficient support for RDF inference queries and knowledge management, *Proceedings of the International Database Engineering and Applications Symposium (IDEAS)*, 2009, Cetraro, Italy.
[MCGL10] McGlothlin, J.P. and L. Khan, Materializing and persisting inferred and uncertain knowledge in RDF datasets, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010, Atlanta, GA.
[MCNA07] Mcnabb, A.W., Monson, C.K., and K.D. Seppi, MRPSO: MapReduce particle swarm optimization, *Proceedings of the Annual Conference on Genetic and Evolutionary Computation (GECCO)*, 2007, London, England, UK.
[MIKA08] Mika, P. and G. Tummarello, Web semantics in the clouds, *IEEE Intelligent Systems*, 23(5), 82–87, Sept./Oct. 2008.
[MORE07] Moreira, J.E., M.M. Michael, D.D. Silva, D. Shiloach, P. Dube, and L. Zhang, Scalability of the Nutch search engine, *Proceedings of the 21st Annual International Conference on Supercomputing (ICS '07)*, pp. 3–12, June 2007, Rotterdam, The Netherlands.
[MORE08] Moretti, C., K. Steinhaeuser, D. Thain, and N. Chawla, Scaling up classifiers to cloud computers, *Proceedings of the IEEE International Conference on Data Mining (ICDM '08)*, 2008, Pisa, Italy.
[NEUM08] Neumann, T. and G. Weikum, RDF-3X: A RISC-Style Engine for RDF, *Proceedings of VLDB Endowment*, 1(1), 647–659, 2008.
[NEWM08] Newman, A., J. Hunter, Y.F. Li, C. Bouton, and M. Davis, A scale-out RDF molecule store for distributed processing of biomedical data, *Proceedings of the Semantic Web for Health Care and Life Sciences Workshop*, 2008, Karlsruhe, Germany.
[OLST08] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, Pig latin: A not-so-foreign language for data processing, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008, Vancouver, BC, Canada.
[ONTO] Ontotext AD: http://www.ontotext.com/owlim/big/index.html.

[OPEN] OpenRDF: http://www.openrdf.org.

[PANT07] Pantel, P. Data catalysis: Facilitating large-scale natural language data processing, *Proceedings of the International Symposium on Universal Communication*, 2007, Kuoto, Japan.

[ROHL07] Rohloff, K., M. Dean, I. Emmons, D. Ryder, and J. Sumner, An evaluation of triple-store technologies for large data stores, *Proceedings of the OTM Confederated International Conference on the Move to Meaningful Internet Systems*, 2007, Vilamoura, Portugal.

[SCHM09] Schmidt, M., T. Hornung, G. Lausen, and C. Pinkel, SP2Bench: A SPARQL performance benchmark, *Proceedings of the 25th International Conference on Data Engineering (ICDE '09)*, 2009, Shanghai, China.

[SIDI08] Sidirourgos, L., R. Goncalves, M. Kersten, N. Nes, and S. Manegold, Column-store support for RDF data management: Not all swans are white, *Proceedings of VLDB Endowment*, 1(2): 1553–1563, August 2008.

[SISM10] Sismanis, Y., S. Das, R. Gemulla, P. Haas, K. Beyer, and J. McPherson, Ricardo: Integrating R and Hadoop, *Proceedings of the ACM SIGMOD International Conference on the Management of Data (SIGMOD)*, 2010, Indianapolis, IN.

[STOC08] Stocker, M., A. Seaborne, A. Bernstein, C. Kiefer, and D. Reynolds, SPARQL basic graph pattern optimization using selectivity estimation, *WWW '08: Proceedings of the 17th International Conference on World Wide Web*, 2008, Beijing, China.

[STON05] Stonebraker, M., D. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau et al., C-store: A column-oriented DBMS, *VLDB '05: Proceedings of the 31st International Conference on Very Large Data Bases*, pp. 553–564, 2005, Trondheim, Norway.

[URBA09] Urbani, J., S. Kotoulas, E. Oren, and F. van Harmelen, Scalable distributed reasoning using MapReduce, *Proceedings of the International Semantic Web Conference*, 2009, Linz, Austria.

[W3a] World Wide Web Consortium: http://www.w3.org/2001/sw/RDFCore/ntriples.

[W3b] World Wide Web Consortium: http://www.w3.org/TR/rdf-concepts/#dfn-rdf-triple.

[W3c] World Wide Web Consortium: http://www.w3.org/TR/rdf-primer.

[W3d] World Wide Web Consortium: http://www.w3.org/TR/rdf-sparql-query/#defn_QueryVariable.

[W3e] World Wide Web Consortium: http://www.w3.org/TR/rdf-sparql-query/#defn_TriplePattern.

[W3f] World Wide Web Consortium: http://www.w3.org/TR/rdf-syntax-grammar.

[WANG10] Wang, J., S. Wu, H. Gao, J. Li, and B.C. Ooi, Indexing multi-dimensional data in a cloud system, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2010, Indianapolis, IN.

[WEAV09] Weaver, J. and J.A. Hendler, Parallel materialization of the finite RDFS closure for hundreds of millions of triples, *Proceedings of the Eighth International Semantic Web Conference*, 2009, Linz, Austria.

[WEIS08] Weiss, C., P. Karras, and A. Bernstein, Hexastore: Sextuple indexing for semantic web data management, *Proceedings of VLDB Endowment*, 1(1): 1008–1019, 2008.

[WIKIa] http://wiki.apache.org/hadoop/JobTracker.

[WIKIb] http://wiki.apache.org/hadoop/TaskTracker.

[XU10] Xu, Y., P. Kostamaa, and L. Gao, Integrating Hadoop and parallel DBMs, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2010, Indianapolis, IN.

第 14 章　云端社交网络

14.1　概述

本章介绍我们正在开发的基于云的系统 SNODSOC（Stream－based Novel class Detection for Social network analysis），基于流的新颖的社会网络分析检测，用于检测社交博客中不断发展的模式和趋势。SNODSOC 扩展了强大的数据挖掘系统，称为 SNOD（基于流的新颖类检测 Stream－based Novel class Detection，SNOD），用于检测博客类。我们还描述了 SNODSOC ＋＋，它是用于检测多个小类的 SNODSOC 的扩展版本。

社交媒体如 Facebook、Twitter 和 YouTube 已成为群体中最流行的沟通方式，可以彼此分享信息。社交媒体传播在许多方面与传统的数据传播不同。例如，通过社交媒体通信，可以在很短的时间内交换多个消息。此外，通信消息（例如，博客和推文）经常缩写，难以理解。为了理解各种社交媒体环境下群体的动机、情感和行为，其中有些是恶意的消息，我们需要使用工具，来理解那些通常表示为图形的社交网络通信消息。为了解决这个需要，我们设计了一个用于分析基于流的社交媒体通信数据的语义框架。

我们开发了一种强大的机器学习工具，称为 SNOD。SNOD 是一种独特的数据流分类技术，可以对数据流中的新类进行分类和检测。SNOD 已经成功应用于 NASA 的航空安全报告系统（Aviation Safety Reporting System，ASRS）的数据集。SNOD 也有许多潜在的应用，如分析社交网络、信用卡欺诈检测、博客和推文检测以及文本流分类。我们利用 SNOD 开发了一个名为 SNODSOC 的复杂社交网络分析系统。要挖掘博客和 Twitter 消息为数据流的分类问题建模，需要 SNODSOC 分析社交网络数据，如博客和 Twitter 消息。我们正在使用云计算框架来实施 SNODSOC。由于需要处理大量社交媒体数据，因此需要基于云的实施来保证可扩展性和良好的性能。

除了 SNODSOC，我们还开发了位置提取（LOCEXT）、概念/实体提取（ENTEXT）和本体构建（ONTCON）的工具。这些工具正在被整合，以开发用于分析社交媒体通信数据的语义框架。集成系统称为 SNODSOC ＋＋。图 14.1 说明了我们的框架。

本章结构如下：14.2 节讨论用于开发 SNODSOC、LOCEXT、ENTEXT 和 ONTCON 可扩展解决方案的基础技术，这些技术包括 SNOD，以及用于位置提取、实体提取、本体构建以及云查询处理的基本工具；SNODSOC 将在 14.3 节中讨论；SNODSOC ＋＋将在 14.4 节中讨论；14.5 节讨论基于云的社交网络分析（Social Network Analysis，SNA）；14.6 节讨论相关工作；14.7 节得出结论。图 14.2 说明了本章的内容。

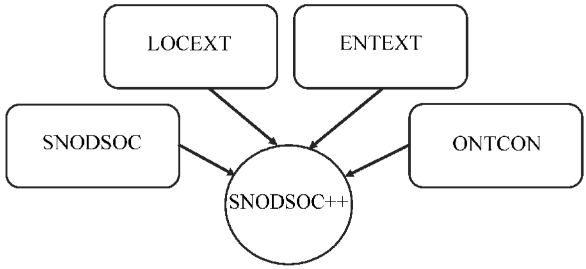


图 14.1 框架（内容来自 Abrol，S.，Khan，L.，Khadilkar，V.，Cadenhead，T. Design and implementation of SNODSOC：Novel class detection for social network analysis，Proceedings of the 2012 International Conference on Intelligence and Security Informatics，p. 215 – 220. © (2012) IEEE. 已获得授权）

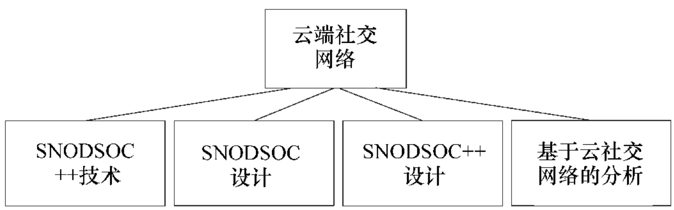


图 14.2 云端社交网络（内容来自 Abrol，S.，Khan，L.，Khadilkar，V.，Cadenhead，T. Design and implementation of SNODSOC：Novel class detection for social network analysis，Proceedings of the 2012 International Conference on Intelligence and Security Informatics，p. 215 – 220. © (2012) IEEE. 已获得授权）

14.2 SNODSOC 和 SNODSOC + + 的基础技术

在本节中，主要说明我们开发的几种工具，它们构成了图 14.1 所示组件的基础。这些组件在 14.3 节中讨论。

14.2.1 SNOD

SNOD 使用我们的数据流学习算法来检测新的类（见本章参考文献 [MASU11a]、[MASU11b]、[MASU12]、[MASU10]）。数据流分类中的三个主要挑战是无限流长度、概念漂移和概念演化。SNOD 通过应用混合批量增加过程，来解决无限流长度和概念漂移问题，该过程执行方式如下：数据流被分成等大小的块，从每个块训练出对应的一个分类模型。

使用 L 个这样的模型的集合来对未标记的数据进行分类。当从一个数据块中训练一个新的模型时，它将替换该集合中一个现有的模型。以这种方式，集合将保持最新状态。通过维持固定大小的集合来解决无限流长度问题，并且通过保持集合的实时性来解决概念漂移问题。SNOD 通过自动检测数据流中的新类来解决概念演化问题。为了检测一个新类，它首先在训练数据周围建立一个决策边界。在未标记数据的分类过程中，首先识别出决策边界外的测试数据点。这样的数据点称为过滤异常值（F 异常值），它们

表示与训练数据分离的数据点。如果发现足够多数量的 F 异常值在它们之间表现出强烈的凝聚力（即它们靠近在一起），则 F 异常值被分类为新的类别实例。

图 14.3 总结了本章参考文献［MASU11a］提出的 SNOD 算法。分类模型是从最后一个标记的数据块中进行训练的。此模型用于更新现有集合。流中的最新数据点则集成测试。如果发现它是一个异常值，它被暂时存储在缓冲区内。否则，它将立即使用当前的集合分类。临时缓冲区被定期处理，以检测缓冲区中的实例是否属于新类。

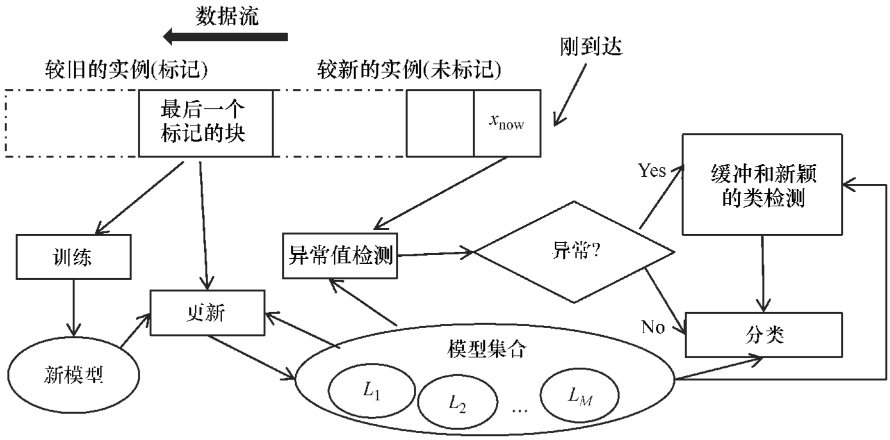


图 14.3　SNOD 算法

14.2.2　位置提取

这里将讨论我们以前在位置提取和社交网络分析方面的一些工作。我们的第一个相关工作是 MapIt（见本章参考文献［ABRO09］），使用高效的算法和启发式算法来识别 Craigslist 广告中存在的非结构化文本中的正确位置，并消除歧义。与地理解析（确定在非结构化内容中出现的文本单词和短语的地理坐标）相关的主要挑战是解决歧义问题。这里存在两种歧义：地理/非地理和地理/地理模棱两可性。地理/非地理歧义是具有另一个非地理意义地名的情况，例如，巴黎可能是法国的首府，或者可能指的是社会名流场所巴黎希尔顿。地理位置/地理模棱两可性，出现在具有相同名称但不同地理位置的两者之间。例如，巴黎是法国的首府，也是得克萨斯州的一个城市。史密斯等人在报告中说，在其语料库中发生的所有名称中有 92% 是不明确的。MapIt 通过使用九点启发式算法来解决这种模棱两可性，能以 85% 的准确度识别定位到街道级别，这超过了以前的相关工作。我们开发了一个功能完整的原型，并对从 Craigslist 网站收集的实际数据集进行了测试。

在我们的第二个相关工作（见本章参考文献［CHAN11］）中，使用概率框架（PDF）来估计每个用户的城市级 Twitter 用户的位置。这些概率基于 Twitter 消息的内容，主要是借助 Twitter 社交网络中不同用户之间交互产生的 tweet 回复消息。使用 tweet 回复消息提供了与用户位置相关联的词语，从而减少了该地理名字在空间分布中的干扰。我们还为每个用户提供了最可能的若干城市名单。我们发现，使用类似的 PDF，与

以前精度约为10%的工作相比较，我们对100英里以内实际用户位置的定位，定位时间是原来的22%。

14.2.3 实体/概念提取和集成

我们开发了一种机器学习方法来提取实体和关系。特别是，我们对现有机器学习方法的两种扩展进行了调查研究。

14.2.3.1 语言扩展

为了抽取实体和关系，需要将其他的语言知识来源纳入到现有的机器学习框架。首先，我们研究了信息是如何从固定本体派生的（并且可以与学习分类器集成以进行分类决策）。我们使用这样的信息作为训练分类器的特征数据，还直接使用可用的本体对实体或关系进行分类，只有本体不存在所需信息的情况下才使用学习分类器。

14.2.3.2 语言以外的扩展

虽然监督机器学习方法减少了系统对本体的依赖，但是基于学习系统的性能在很大程度上取决于大量标记数据的可用性，这可能是昂贵的。因此，我们开发了三种语言以外的扩展，旨在提高提取系统在有限的标记数据方面的健壮性。它们是：①将基于知识和学习的方法与实体提取和关系提取相结合；②结合域适应技术；③自动生成附加的训练数据。

14.2.3.3 实体整合

一旦实体和关系从自由文本中提取出来，就必须将它们整合在一起，以创建一个组合的语义表示。在实体提取期间，一些被提取的实体可能是错误的或者不相关的，一些被提取的实体也有可能是相关的。因此，我们将抛弃不相干的实体，并保留相关的实体。接下来，我们就整合来自多个文件的实体。在不同文献的多个参考信息中，实体可能指向现实生活中的同一实体。我们解决了以下两个相关的挑战：①消除由信息提取技术产生的文档中的不相关实体；②将提取的实体合并到文档中，以便在确定相同的实体之间建立“相同”的实体关系。

14.2.4 本体构建

基于本章参考文献［KHAN02］中描述的工作，我们正在开发一种用于自动构建域依赖本体的具有很大潜力的新方法。方法的关键是开发层次结构，以及层次结构中每个节点中通用本体的概念选择。对于开发层次结构，我们修改了从上到下构建层次结构的现有自组织树算法（Self Organizing Tree Algorithm，SOTA）。

接下来，需要为层次结构中的每个节点分配一个概念。为此，我们部署两种策略，并采用自下而上的概念分配机制。首先，对于由一组文档组成的每个群集，我们基于修改的用于主题跟踪的Rocchio算法分配一个主题。然而，如果有多个概念是主题的候选者，我们就提出一种智能的仲裁方法。接下来，为了把概念分配给层次结构中的内部节点，我们使用语言学本体的WordNet。内部节点的派生概念也将在WordNet中被识别。从这些已识别的概念和它们的上位词，我们就可以确定一个更通用的概念，也可以被分配为内部节点（见本章参考文献［MCGL10］、［ALIP10］）的概念。

14.2.5 云查询处理

如第 13 章所述，我们在云上开发了一个 SPARQL 查询处理系统。实际上，我们已经开发了一个框架通过 Hadoop 查询 RDF 数据存储，使用 Pellet 推理器在不同的阶段进行推理。具体实现时，使用与 Hadoop 的 Map/Reduce 功能相结合的 Pellet 库进行实时查询推理。我们的 RDF 查询处理由两个主要步骤组成：预处理以及查询执行和优化。

14.2.5.1 预处理

要对 RDF 数据执行 SPARQL 查询，就需要对数据执行预处理，并将预处理数据存储在 HDFS 中。通过编写一个单独的 Map/Reduce 任务来执行，将 RDF/XML 数据转换为 *N* - Triples，同时也是为了前缀生成。我们的存储策略是基于谓词分割（见本章参考文献［HUSAIN11］）。

14.2.5.2 查询执行和优化

我们为 Hadoop 开发了一个 SPARQL 查询执行和优化模块。由于我们的存储策略是基于谓词拆分，首先我们要检查查询中存在的谓词，其次我们检查与谓词匹配的输入文件的子集，最后 SPARQL 查询通常在其中具有许多连接，并且所有这些连接可能无法在单个 Map/Reduce 作业中执行。因此，我们已经开发出一种算法来决定每种查询所需的作业数量。作为优化的一部分，我们采用贪婪策略和成本优化来减少查询处理时间（见本章参考文献［HUSAIN11］）。

14.3 SNODSOC 设计

14.3.1 模块概述

互联网社交媒体服务（例，微博和社交网络），往往由类似 Twitter 等这样的平台所提供，这些服务都在用户群中大幅增长。其中，微博现象早在 2000 年代中期就被讨论过，我们能看到用户使用这个服务，来谈论他们的日常活动，并寻求或分享信息。这种增长激发了人们对使用这些平台提供的数据来提取各种类型信息的兴趣。我们设计了以下社交网络分析工具（如图 14.1 所示）。

1）趋势分析，包括 SNODSOC。

2）从博客和推文中提取用户的地理位置（LOCEXT）。

3）消息分类和实体提取（ENTEXT）。

4）本体构建（ONTCON）。

这些工具将构成 SNODSOC ++框架，可以进行复杂的社交网络分析。特别是，我们开发了一套技术，将有助于新兴趋势分析、地理位置提取、消息分类和实体提取和本体构建。这里所提到的提取过程将是非常耗时的。因此，我们正在利用云计算解决方案来加快提取过程。提取的知识可用于为用户提供个性化服务，如本地新闻、本地广告、应用共享等。Twitter 在不同的地理位置拥有超过 2 亿个账户，短信或推文可形成一个巨

大的数据集，可以分析这些数据去提取地理信息。

14.3.2　SNODSOC 和趋势分析

Twitter 不仅成为社交网络的主要形式，而且也成为一种新的、不断增长的沟通方式。Twitter 的用户数量和 Tweet 的数量在继续增加。Tweet 能很快反映出突发的新闻故事。通常，关于自然灾害、犯罪或政治事件的第一个消息来源是 Tweet。Twitter 也是团体和社区组织的沟通机制。Twitter 对政治事件和抗议团体产生了重大影响。

Twitter 可以为个人提供一种沟通方式，而不依赖传统媒体。通过 Twitter 服务的这种新的交流模式让流数据挖掘面临新的挑战。Tweet 在连续流中流动。我们的目标是破译和监控这些推文中的主题，并检测何时出现趋势。这包括比如运动或时尚等主题的一般变化，它包括新的、快速出现的趋势，如死亡或灾难。将 Twitter 消息与趋势和主题正确关联是一个具有挑战性的难题。由于输入的消息是连续和大量的，这些挑战最好由流式传输模式来解决。

我们系统的核心是 SNOD 的分类模型（见图 14.4）。为了构建这个模型，我们可以使用 K－NN 方法或者多标签文本分类方法。将使用从新的 Twitter 消息中获取的特征数据来逐步更新此模型。当选择新的消息（即，推文）作为训练示例时，将对其进行分析并记录其特征。该记录的数据将被存储在一个临时的数据库中，该数据库将一次保存一批 N 条消息的数据。当一批数据被处理后，它将被丢弃，再继续存储新的批次。每批数据都将进行特征提取和选择。所选择的特征将用于计算特征向量（每个消息有一个向量），这些向量被用于使用增量学习技术来更新现有的分类模型。当系统中出现未知的可执行文件时，首先，将监视和记录其运行时间和网络行为。然后，该数据将经历类似的特征提取阶段，但是由于在训练/更新阶段中选择的相同特征被用于构建特征向量，所以不

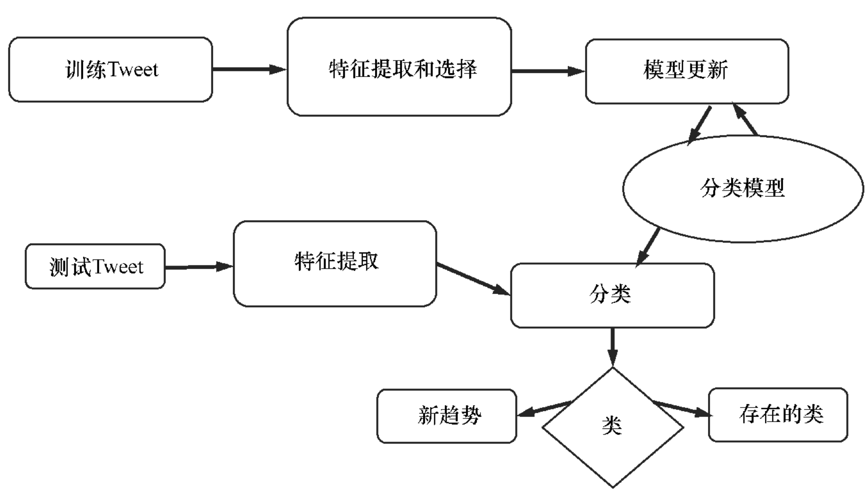


图 14.4　分类模型

需要特征选择。该特征向量将使用分类模型进行分类。根据分类标准，采取相应的措施。

14.3.2.1 新类检测

上述特征提取和分类过程是为固定大小的训练数据生成特征向量。为了应对日益增长的 Tweet，我们必须将其扩展到数据流框架。我们的数据流分类器（见本章参考文献 [MASU11a]、[MASU10]、[MASU11b]、[MASU12]）可以处理大量的训练和测试数据，也能处理概念漂移、新类和流中的特征演进。回想一下，我们假设数据流被划分成相等大小的块。该系统的核心是 M 分类器 $\{L_1, \cdots, L_M\}$ 的集合 L。当新的未标记的可执行文件（测试实例）到达时，集合用于对实例进行分类。如果测试实例被识别为异常值，则将其临时存储在缓冲区 buf 中以供进一步检查。否则，如果它不是异常的，那么它被归类为良性或恶意。定期检查缓冲区，看看是否出现了一个新类。如果检测到一个新类，则属于新类的实例被相应地识别和标记。一旦新的标签数据块到达，它被用于训练分类器 L'。然后，通过根据最新的标记数据块 D_n 的精度选择 $M+1$ 分类器 $L \cup \{L'\}$ 中最好的 M 个分类器来更新现有的集合（参见算法 A）。

新类检测技术的核心概念是属于一个共同类的数据点应该让它们彼此更接近（内聚），也应该与属于其他类（分离）的数据点相差更大些。当对新类进行检查时，我们在 buf 中的异常值之间寻找强烈的凝聚力，以及异常值与训练数据之间较大的分离性。如果发现这种强烈的凝聚力和分离性，我们将声明这是一个新类。当人类专家揭露新类的真实标签时，将这些实例用作训练数据，并对新模型进行训练。最后，使用该模型更新现有集合。因此，模型集合的组合，将不断丰富新类。算法 A 总结了该技术。

在训练过程中要创建决策边界：使用 K – means 对训练数据进行聚类，并将每个簇的汇总保存为伪点。然后丢弃原始训练数据。这些伪点构成训练数据的决策边界。

14.3.2.2 存储簇汇总信息

对于每个簇，我们将以下汇总信息存储在伪点数据结构中：①权重 w（簇中的总点数）；②中心 μ；③半径 r（质心和簇中最远的数据点之间的距离）；④平均距离 μd（从每个点到簇中心的平均距离）。因此，符号 $w(h)$ 表示伪点 h 的权重值，依此类推。在计算簇汇总之后，原始数据被丢弃，只有伪点存储在内存中。任何具有太少（少于三个）实例的伪点都被认为是干扰并且将被消除。因此，用于存储训练数据的存储器大小就变得恒定了，即，$O(K)$。

异常值检测和过滤：每个伪点 h 对应于特征空间中的超球体，这个超球体具有中心 $\mu(h)$ 和半径 $R(h)$。由伪点 h 覆盖特征空间的部分是假点区域 $RE(h)$。分类器 L_i 中所有伪点区域的并集是 L_i 中训练数据的决策边界。分类器 L 集合的决策边界是集合中所有分类器的决策边界的并集。决策边界在新类检测中起着重要的作用。它定义了现有类中实例的物理边界。

每个测试实例都经过测试，以查看它是否在现有类实例的决策边界之外。为了计算是否越界，我们从每个分类器的测试实例中找到最近的伪点。如果测试实例在所有分类

器中每个伪点的半径之外，则被认为是异常值。我们将决策边界外的任何测试实例称为 F 异常值。

算法 A：基于流的新类检测

输入 L：最佳 M 分类器的当前集合

x：一个未分类的未知实例

输出：x 的类标签或新类的检测

1. outlier←detectOutlier（x，L）

2. if（outlier=，false）then

3. classify（x，L）//分类为良性或恶意

4. else buf⇐x//保存在异常值缓冲区

5. if time to check buf then

found←DetectNovelClass（buf，L）

6. if found then {Y←Novel_instances（D_n）;} //标记新类实例

//如果有一个新的标签块 D_n 已经准备好进行训练

7. L'←Train－new－model（D_n）

8. L←Update（L，L□，D_n）

新类检测：我们对 F 异常值进行多个计算，以检测新类是否产生。对于每个 F 异常值 x，我们将其 λ_c 邻域 $\lambda_c(x)$ 定义为属于类 c 的 x 的最近 N 个邻居的集合。这里，N 是用户定义的参数。例如，邻域 $\lambda_+(x)$ 是类 c_+ 与 x 的最近 N 个实例的集合。类似地，$\lambda_o(x)$ 是指最接近 x 的 N 个 F 异常值的集合。

使用此邻域信息，我们计算 N 近邻剪影系数（N－NSC），度量如下：让 $a(x)$ 是从 f 异常值 x 到 $\lambda_o(x)$ 的实例的平均距离，让 $b_c(x)$ 是从 x 到 $\lambda_c(x)$ 实例的平均距离（其中 c 是一个现有类）。让 $b_{\min}(x)$ 为所有类 c 的最小 $b_c(x)$。然后，公式 $N-\mathrm{NSC}(x)$ 由下式给出：

$$N-\mathrm{NSC}(x)=\frac{b_{\min}(x)-a(x)}{\max[b_{\min}(x),a(x)]}$$

根据这个定义，N－NSC 的值在 −1 和 +1 之间。它实际上是聚合与分离的统一测量。负值表示 x 靠近其他类（较少分离），并且远离其自身的类（较弱的内聚力）。如果对于所有的分类器，至少有 $N'(>N)$ 个 F 异常值，其 N－NSC 为正，那么我们就可以声明一个新类。

值得注意的是，N 的值越大，我们可以决定一个新类是否已经到来的信心就越大。但是，如果 N 太大，那么在相应的数据区块中属于新类的实例总数是 $\leqslant N$ 的，我们也可能无法检测到一个新的类。因此，通过对现实数据集的实验测试，确定 N 的最优值是很重要的。

具有特征演化的新类检测：表示流中数据点的特征空间可能会随时间变化而变化，我们称之为“特征演化”现象（见本章参考文献［LIN11，GOYA09］）。例如，考虑一个文本流，其中每个数据点都是一条 Tweet 消息，每个单词都是一个特征。由于不可能

知道将来会出现哪些词语，所以完整的特征空间是未知的。此外，通常只使用一部分词作为特征集，因为大多数单词可能对于分类是冗余的。因此，在任何给定时间、特征空间都由使用某些选择标准的有用单词（即，特征）来定义。在未来，新词可能成为有用的，旧的有用词可能就变得多余，特征空间动态地改变。为了应对特征演化，分类模型应该能够正确地分类具有不同特征空间的数据点，而不是模型的特征空间。下面的示例说明两个连续数据区块中的特征演化。在第 i 块，关键特征集是｛跑道，爬坡｝和（$i+1$）区块，关键特征集是｛跑道，空隙，坡道｝。显然，两个不同区块中的关键特征集是不同的，而新的关键特征随新类出现在（$i+1$）区块中。如果我们使用第 i 块的特征集测试（$i+1$）块的实例，我们可能无法检测到这个新类。

大多数现有的数据流分类技术都假设流中数据点的特征空间是静态的。如上例所示，对于某些数据流（如文本流），此假设可能不切实际。我们的技术将考虑特征空间的动态特性，并为特征空间动态的分类和新类检测提供一个合适的解决方案。如果特征空间是动态的，那么我们将在不同的数据区块中拥有不同的特征集。因此，在集合中的每个模型都将在不同的特征集中被训练。此外，测试实例的特征空间也不同于模型的特征空间。当我们需要对实例进行分类时，我们需要为模型和测试实例提供一个同构的特征空间。有三种可能的替代方法：①有损固定转换（或简称为 Lossy－F 转换）：在这里，我们对整个流使用相同的特征集，实际上就是为第一个数据区块（或前 n 个数据区块）选择的特征集。这将使特征集固定，因此流中的所有实例（无论是培训还是测试）都将映射到此特征集中。我们称之为有损转换，是因为将来的模型和实例可能会因转换而失去重要的特性。②有损本地转换（或简称为 Lossy－L 转换）：在这种情况下，每个训练区块以及从区块建立的模型，都将使用特征抽取和选择技术来选择自己的特征集。当将测试实例使用一个模型 M_i 进行分类时，模型将使用它自己的特征集作为测试实例的特征集。此转换也是有缺点的，因为该转换的测试实例可能会丢失重要的特性。③无损均质转换（或简称无损转换）：在这里，每个模型都有自己选择的一组特征。当测试实例 x 将使用一个模型 M_i 分类时，模型和实例都将其特征集转换为其特征集的联合。由于模型和测试实例都保留了它们的维度（即特征），而转换的特征空间对于模型和测试实例都是同类的，因此我们称之为“无损耗均质”。这是因为在转换的过程中没有任何有用的特征丢失。由于实例可能没有固定的、预先确定的特征向量，并且由于文本，我们得到可变长度的高维特征。因此，我们应用特征选择来选出每个区块的最佳特征，并应用上述技术来构造同类的特征空间。为了解决稀疏问题，我们正在研究相关可能的技术。为了跟踪带有短消息的连续流当中的主题/趋势，可以根据前景模型和背景模型的组合来检查混合模型。前景模型将根据上文所述的基于集合技术捕捉“现在发生的事情”，背景模型将对抗数据稀疏性（见本章参考文献［LIN11］）。

14.3.3 内容驱动的位置提取

Twitter 允许用户将其地理位置指定为用户信息（元数据）。此位置信息由用户手动输入或启用 GPS 的设备进行更新。启用 GPS 的设备更新用户位置的功能尚未被大量用

户采用（见本章参考文献［CHEN10］）。因此，大多数用户的地理位置数据可能丢失或不正确。依赖用户手动更新位置有几个缺点。

1）用户可能有不正确的地理位置数据。例如，Twitter 用户可以输入他/她的位置为“Krypton”。这可能不是一个真正地理位置的名称。

2）用户不一定总是有城市级的位置。用户可以含糊地输入位置名称、城市名称（如亚利桑那州）或国家名称（如美国）。这些位置名称不能直接用于用户确定城市级别的位置。

3）用户可能有多个位置。如果用户移动到不同的位置，他/她可能会在他/她的 Twitter 页面的元数据中提到不止一个位置。这使得很难确定其目前的单一城市级地理位置。

4）用户可能有不完整的位置数据。用户可能指定了一个模棱两可的名称，可能引用不同的位置。例如，如果用户指定了如“华盛顿”之类的位置，则该名称可以与一个国家名称或城市名称（华盛顿特区）相关。这些不确定类型的名称使得我们很难确定用户的准确位置。

因此，这种情况下，确定一个用户城市级地理位置的数据的可靠性很低。为了克服用户稀少的地理定位信息的问题，我们仅仅基于他/她的 Tweet 内容以及相关回复消息的内容，来评估 Twitter 用户的城市级地理位置。我们使用一个 PDF，它考虑了一个特定会话的 Tweet 消息中术语的分布，这个会话由用户发起。并包含 tweet 的回复消息。我们的工具建立在 14.2 节讨论的基础工作上。

14.3.3.1　动机

在 Twitter 上，用户可以发布被称为 Tweet（推文）的微博，可供其他用户阅读。随着这种微博服务，Twitter 还提供了一个社交网络服务，用户（关注者）可以“关注”另一个用户（被关注者）。社会网络的每个边都由这种“关注”关系形成。作为关注者，用户收到所有关注者发布的推文，反过来可以回复这些推文。这个回复推文由被关注者的关注者收到。这形成了两个不同用户之间基础的对话。Huberman 等人（见本章参考文献［HUBE09］）分析了超过30 万的用户，发现 reply – tweet 和定向推文占 Twitter 的所有帖子的 25.4%。这表明在 Twitter 用户之间广泛使用了 reply – tweet 功能。（图 14.5 举例说明了一个例子）

我们的直觉告诉我们，用户之间的对话可以与一系列主题相关，例如天气、体育等，包括某些特定位置的主题，例如与城市相关的事件或对特定地点的引用，或城市中一个实体。我们假设这一组话题在对话中保持不变。当用户发布推文消息时，可以将其视为对话的开始。当另一个用户向原始的推文发帖回复推文时，此会话可以继续。没有详细说明回复的主题，可以假设主题与原始的推文消息相同。在这种假设下，答复推文的任何内容都可以与原始的推文信息相关。例如，考虑图 14.5 中两个用户交换的推文消息。一个用户发推文消息，另一个用户回复推文消息。请注意，会话中的会话主题保持不变。因此，通过结合上述假设，使用可能具有位置特定数据的推文内容，我们应该获得比如果我们孤立地考虑推文或者我们只依赖于用户指定位置的更好的结果。



图 14.5　推文示例

14.3.3.2　挑战：拟议方案

仅仅使用推文内容来估计 Twitter 用户的位置，以及上面提及的直觉，都将给我们带来一些挑战。这些挑战是基于推文中使用自然语言的语义复杂性。有些用户可能会在其推文中使用非标准词汇。一个城市的用户可以使用不同的名字指向相同位置的特定实体。例如，来自洛杉矶的用户可以将城市的名字称为洛杉矶、洛杉矶市、洛杉矶城市等。用户也可以引用具有相同名称的不同位置。例如，来自纽约的用户可以将纽约州的街道名字称为第六街，奥斯汀的用户可以在奥斯丁引用相同名称的街道。这些例子泛化了这些术语的空间分布。

推文并不总是包含特定位置的术语。他们可能会包含许多来自自然语言的一般词语，因为许多用户对一般主题发表推文。因此，推文的内容被认为是嘈杂的。

推文可能包含指向多个位置的术语，这会降低推算用户特定位置的能力。作为假设，在考虑对话时，会话的主题可能在整个会话中保持不变。在回复推文中更改主题可能会导致多个位置特定术语或术语空间被宽泛分布。对于这些问题，我们提出了两种方法，从他的消息或博客中提取用户的城市居住地点。

在第一种方法中，我们使用 PDF 来估计每个 Twitter 用户的城市级位置。这种可能性是基于 Twitter 消息的内容，借助 Twitter 社交网络中不同用户之间交互产生的回复推文消息。使用回复的推文消息可以更好地提供与用户位置相关联的字词，从而减少术语空间分布中的干扰。我们还为每个用户提供了最可能的城市名单。

如前所述，仅从推文内容中提取地理位置特定信息，很具有挑战性。使用社交交互模型，我们用用户之间任何交互中所使用的推文内容，来确定对话期间使用术语的概率分布。在第二种方法中，我们研究以用户为中心的位置挖掘方法，该方法希望从他的消

息中识别特定用户的单个城市级家庭位置。与第一种方法不同，第一种方法是针对来自社交互动的消息，这种方法是一种更为简单的方法，适用于所有社交媒体，包括 Twitter、博客等。

社会交往概率分布模型（Probability Distribution Model，PDM）：在这种概率分布中，每个用户都可以被认为属于一个特定的城市。因此，用户的推文可以被假设为与特定城市相关，特别是发布推文的用户所在的地理位置，也就是说，用户推文中出现的术语可被分配为与用户城市相关的术语。这就形成了在完整数据集中考虑一组城市术语的基本分布。对于每个城市 c，整个数据集中项 t 的概率分布被给出为

$$p(t \mid c) = |\{t \mid t \in \text{terms} \wedge t \text{ occurs in city } c\}| / |t|$$

形成大小为 $n \times m$ 的概率分布矩阵，其中 n 是项列表的大小，即字典的大小，m 是被考虑用于评估的数据集中的城市总数。

基于回复的概率分布模型（Reply－Based Probability Distribution Model，RBPDM）：在 PDM 的基本概率分布计算中，用户在其推文中使用的术语被分配给用户所属的城市。它不考虑不同推文消息之间的关系。

Twitter 提供了一个功能，可以在称为回复标签的推文中标记另一个用户。该标签将消息引导到在推文中寻址的用户。考虑到这一点，推文信息可以分为三种不同的类型：

1）推文消息是用户通常在 Twitter 上发布的一般消息。这些推文消息不包含任何回复标签。这种类型的推文消息中使用的术语，可以用于与用户城市术语的空间分布进行直接关联。

2）推文消息是包含用户回复标签的消息。这种类型的消息称为推文回复消息。此消息通常用于回复另一个用户发布的某个推文。推文回复消息将被发送给正在被回复的用户，也就是发给原始推文消息的用户。此推文通常会在推文消息开头包含回复标签。

3）推文消息是具有回复标签的消息，但可能不是回复推文。它可能是一个发给用户的推文消息，但不需要该用户对该推文回复。此消息可能包含推文单词之间的回复标签。它也可能是用户转发用户的推文消息的重新推送，以便他/她的关注者可以收到推文消息。

当考虑到推文中的回复标签时，会产生两条推文消息之间的关联。回复推文将与来自用户生成回复推送消息的原始推文直接关联。PDM 忽略了推文消息之间的所有关系。在这里，我们考虑不同推文消息之间的这种关系，同时计算数据集中术语的概率分布。这种关系形成了不同用户之间对话的基础，即一个推文消息，其回复消息可以被认为是用户之间的对话。因此，通过假设，话题的主题在回复消息中保持不变，会话中使用的术语可以与会话的主题相关。对话可能涉及与该主题相关的特定于位置的术语，而不是明确地将发短信使用的术语分配给发布推文的用户，可以将完整对话中出现的术语分配给发起对话的用户，因为发起者可以发起涉及他/她地理位置的对话主题位置。因此，当在数据集中遇到回复推文时，我们将推文中涉及的术语分配给发送推文的用户，而不是发布回复推送消息的用户。通过把这些术语分配给不同用户，在不考虑识别不同类型的推文消息及其关系时，我们评估其概率分布。因此，在估计 Twitter 社交网络中用户

的地理位置时，考虑的是网络的社会结构。

术语分布估计：使用在数据集中城市间的术语分布，可以基于最大似然法来计算给定项 t 中城市 c 的概率。

$$p(c|t) = \max_{\forall c \in \text{cities}} p(t|c)$$

用户 u 位于城市 c 中的概率估计,是从城市 c 的用户推文中提取术语的总概率,即

$$p(c|u) = \sum_{(w \in \text{terms})} p(c|w)^{*} p(w)$$

使用该公式，如前所述获得概率估计矩阵，其大小为 $p \times q$，其中 p 是用户列表的大小，q 是在数据集中城市列表的大小。城市级别地理位置的估计可以通过对该用户对应的最高相关城市概率来获得，也可以通过对每个用户的概率估计矩阵进行排序，并从其中列出最上面的 k 个最可能的城市来获得顶级 k 个估计城市的列表。

14.3.3.3 使用地名词典和自然语言处理

在讨论实际运行算法（参见算法 B）之前，我们需要执行数据预处理，最初涉及使用如城市字典之类的外部词典来理解粗俗的口语，并用适当的短语替换它们，以获得语法正确的句子。之后，我们删除了消息中所有没有涉及地理位置的单词（见图 14.6）。

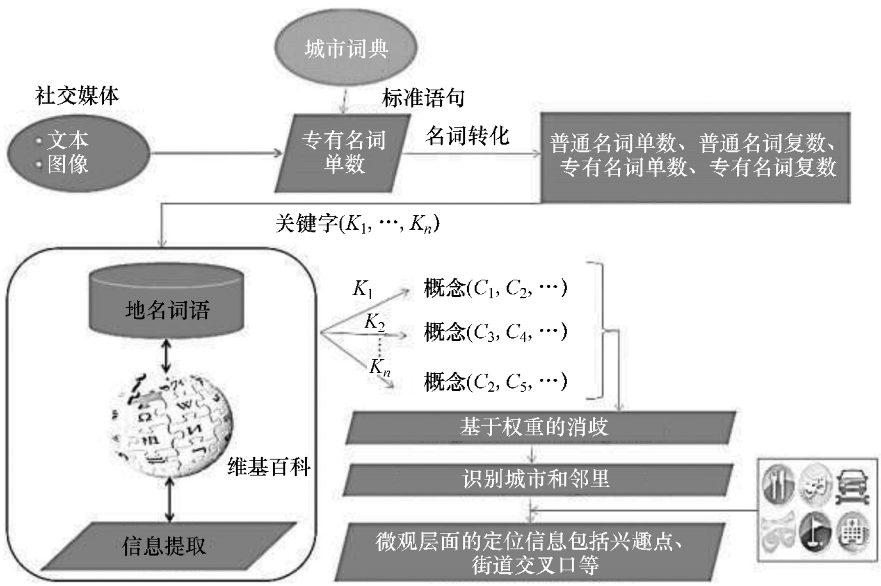


图 14.6　使用地名词典和自然语言处理方法

为此，我们使用英语的部分语音（Part Of Speech，POS）标签。POS 标签从文本中识别所有专有名词，并将其称为关键字 $\{K_1, K_2, \cdots, K_n\}$。在下一步中，搜索 TIGER（拓扑集成地理编码和参考系统，Topologically Integrated Geographic Encoding and Referencing system）数据集，从中识别城市名称。TIGER 数据集是一个开源地名词典，由整个美国的拓扑记录和形状文件组成，具有城市、县、邮政编码、街道段等坐标。

我们搜索 TIGER 地名词典，以了解关于概念 $\{C_1, C_2, \cdots, C_n\}$ 的每个关键字。现在，对于每个关键字，我们的目标是选择列表中的正确概念，换句话说，就是消除位置的歧义。为此，我们使用基于权重的方法来消除歧义。我们根据文本中出现的术语（阶段1）为每个概念分配权重。与一般的概念相比，具体概念被赋予更大的权重。然后，检查概念之间的相关性，其中一个概念包含在另一个概念之中（阶段2）。在这种情况下，具体的概念就从一般的概念中得到提升。如果具体的概念 C_i 是另一个 C_j 的一部分，则 C_j 的权重被加到 C_i 的权重上。

算法 B：位置识别（User_Messages）

输入 UM：用户的所有消息

输出矢量（C，S）：概念和得分矢量

1. for each keyword，K_i//阶段 1
2. for each $C_j \in K_i$//C_j –城市概念
3. for each $T_f \in C_j$
4. type = Type（T_f）
5. If（T_f occurs in UM）then $S_{Cj} = S_{Cj} + S_{\text{type}}$
6. for each K_i//阶段 2
7. for each $C_j \in K_i$
8. for $T_f \in C_j$，$T_s \in C_L$
9. If（$T_f = T_s$）and（$C_j \neq C_L$）then
10. type = Type（T_f）
11. $S_{Sj} = S_{Cj} + S_{\text{type}}$
12. return（C，S）

例如，概念中有城市则得 15 分，有州得 10 分，有国家名字得 5 分。关于“达拉斯”的关键词，考虑到｛City｝达拉斯/｛州｝得克萨斯/｛Country｝美国这个概念。这个概念可得到 15 分，因为达拉斯是一个城市的名字，如果得克萨斯在文中也被提到，它将再获得 10 分。在第一阶段，我们考虑两个关键词之间的关系。考虑到前面的例子，如果｛Dallas，Texas｝是文本中出现的关键词，那么在“达拉斯”中列出的各种概念，将会是｛City｝达拉斯/｛州｝得克萨斯州/｛Country｝美国和其中一个概念“得州”将是｛州｝得州/｛国家｝美国。现在，在第二阶段，我们检查这样的相关概念，其中一个概念包含另一个概念。在这种情况下，更具体的概念从更一般的概念中得到提升。在这里，上述得克萨斯概念提升到更具体的达拉斯概念。在两个阶段完成之后，我们按照权重的降序重新排列概念。接下来，每个概念被赋予取决于它们各自权重的新概率。

一旦确定了城市级位置，该算法将重点关注在识别文本中可能提到的微观层面的位置（博客、推文等）。为此，我们将使用地理空间接近度和基于上下文的消歧算法，该

算法使用丰富的兴趣点（Point Of Interest，POI）数据库（Foursquare Venue 数据集作为一个例子）来识别潜在的微观层面的地方，如咖啡店、学校、餐馆、礼拜场所等。该算法将返回一个地点列表，与该特定关键字匹配，并位于以前由我们已经确定的城市。现在，我们每个关键字的目标是选择列表中的正确概念。换句话说，确定用户所指位置是正确的 POI。为此，我们使用两阶段完成消歧过程。第一阶段由基于上下文的评分组成。地名词典中的每个 POI 条目都与标签相关联，并且通过使用 WordNet 语义相似性度量，我们计算消息和 POI 条目之间的相似性。这是为了根据用户可能参考的地点类型来消除正确 POI 的歧义。之后，在第二阶段中，针对特定用户，我们简单地提升之前确定的 POI 集合中每个 POI 候选者的接近度分数。所以，对于一个一用户，如果在洛杉矶地区有多个星巴克咖啡店的返回信息，那么算法会选择更接近他的舒适区域（很可能是他的家庭或办公室）的星巴克门店。

14.3.4　分类

我们工作的一个主要目标是从推文中提取实体，并对消息进行分类。然后，我们提供隐藏在推文中知识的语义表示。这将使分析师能够更直接地与隐藏的知识进行交互。我们基于 14.2 节讨论的基础工作进行分类。

实体提取：实体是知识表示中的基本元素。我们已经开发了相关技术，来识别消息中几个重要的实体，包括事件、位置、人员、组织等。在一些文献中，已经说明实体提取得到了广泛的研究。最有效的方法是把它看作为分类问题，并应用监督的学习方法。我们也采用相同的策略，并利用许多公开的工具来执行此任务。我们首先使用 FEX（可从 http：//l2r. cs. uiuc. edu/cogcomp/asoftware. php？ skey = FEX 获得）进行特征提取和 BBR（可从 http：//www. stat. rutgers. edu/madigan/BBR/获得）用于逻辑回归，以识别 Twitter 中有趣的实体。我们已成功地使用 FEX 和 BBR 来识别 NASA ASRS 报告（见本章参考文献［AHME10a］）中的名称。

虽然一些现有的算法和工具可供我们使用，但我们必须解决的一个主要挑战是创建标记的数据以方便后面的训练。标记数据的过程通常是开销巨大的。我们提出研究解决这个问题的两个策略：①我们采用自举方法，首先要依靠语言规则，来识别给定类型的实体中那些易于识别的实例，然后使用基于规则的识别器生成监督学习的训练数据；②我们采用一些基于现有工作的域适应技术，来优化存在于其他相关领域（如新闻领域）的训练数据。当我们重用一些现有的训练数据时，这些技术可以有效地避免过度拟合。由于提取的实体目的在于数据挖掘，我们期望可以容忍实体识别中的一些错误。如果我们需要进一步提高准确性，我们将探索手动标记一些示例的可能性。上面讨论的策略可以帮助产生最有前景的积极示例。一般来说，主动学习可以用于选择用于用户判断的例子。

多标签文本分类：社会通信数据和相关信息的分类在这种数据分析中起着重要的作用。为了找到一种非常适合社交媒体数据的分类技术，我们首先需要了解这些数据与其非文本对象的区别。其次，这些数据集中的消息通常采用非正式的写法。这样，消息的

特征，就具有非常高和稀疏的维度，因为这些维度或特征空间，由出现在相应自然语言词汇表中的所有不同词组成，组成情况多种多样。

我们认为的第二个差异在于分类时，其与多个类别关联的趋势越来越大。文本数据集本质上可以是二进制、多类或多重标签。对于二进制类和多类，只能将一个类标签与一个文档相关联，并且类标签关联之间是相互排斥的。然而，在多标签数据的情况下，多个类标签可以同时与一个消息相关联。

在多标签数据分类中，类标签可以同时出现，不同类标签共同出现的频率表示类标签不是彼此独立的。此外，并非所有的类标签组合都出现在数据集中。因此，假设类标签是独立的，但在分类过程中实际上是不正确的。例如，在 ASRS NASA 数据集中，我们共考虑了 21 个类。有些类标签永远不会组合在一起。例如，给出两个属性的标签，（飞机装备问题：关键和较不严重）和（冲突：不那么严重和关键），我们知道两个标签（飞机设备问题：关键）和（冲突：不那么严重）永远不会发生在一起，这是从他们的名字显而易见的。然而，飞机设备问题：严重和冲突，严重级别可以共同发生。在分类过程中，考虑到这种变化共生的概率，可以在一定程度上允许我们生成不共同出现类别标签对的簇。

为了对 Twitter 进行分类，我们利用文本分类。为此，我们使用 SISC 算法（见本章参考文献［AHME09］、［AHME10a］、［AHME10b］），其在少量或有限量的标记训练数据集下，将子空间聚类与 K – NN 方法以及半监督学习方法相结合使用。为了正确解释这些数据的多重标签属性，模糊聚类可以以更有意义的方式执行此解释。事实上，模糊子空间聚类的概念与文本数据的概念匹配，也就是具有高和稀疏的维度以及多标签属性。子空间聚类允许我们在加权超空间（见本章参考文献［FRIG04］）中找到集群，并且可以帮助我们找到只在一个维度的子集中形成簇的文档。SISC（见本章参考文献［AHME09］）是我们在实验中使用的这么一种算法。

14.3.5 本体构建

本体是概念及其相互关系的集合，可以集中提供应用程序域的抽象视图（见本章参考文献［KHAN02］）。基于本体的模型有两个不同的问题/任务：一个是从关键字中提取语义概念，另一个是本体的实际构建。关于第一个问题，关键是确定描述和标识消息的适当概念（如前所述）。这样一来，重要的是确保不相关的概念不会被关联和匹配，相关概念也不会被丢弃。关于第二个问题，就是要自动构建本体。在这里，我们通过提出一种自动构建本体的新方法来共同解决这两个问题。

基于 14.2 节中讨论本体构建的基础工作。我们的方法是以自下而上的方式自动构建本体。为此，我们首先使用一些聚类算法构建层次结构。回想一下，如果文档在内容中彼此相似，则它们将与本体中相同概念相关联。接下来，我们需要为层次结构中的每个节点分配一个概念。如前所述，为此，我们部署了两种策略，并采用自下而上的概念分配机制。首先，对于由一组文档组成的每个群集，我们基于修改的 Rocchio 算法为主题跟踪分配一个主题。然而，如果有多个概念是主题的候选者，我们提出一种智能的仲

裁方法。接下来，为了将概念分配给层次结构中的内部节点，我们使用语言学本体的 WordNet。内部节点的后代概念也将在 WordNet 中被识别。从这些识别的概念和它们的上位词中，我们可以识别一个更为通用的概念，也可以被分配为内部节点的概念。

考虑到是层次结构，我们采用自动化构建本体。为此，需要依靠从上到下构建层次结构的 SOTA。我们修改原始算法，并提出了一种与层次聚类算法相比，以更高的精度构建层次结构的有效算法（见本章参考文献［LUO04］）。为了说明自动本体构建方法的有效性，我们在文本文件中探讨了本体建构。Reuters21578 文本、文档语料库已被使用。我们已经观察到，修改后的 SOTA 在精度方面优于凝聚聚类。这项工作的主要贡献如下：

1）我们提出一种可以用于自动生成本体的新机制，使得我们的方法具有可扩展性。为此，我们修改从上到下构建层次结构的现有 SOTA。

2）此外，为了层次结构中的每个节点都能找到一个适当的概念，我们从 WordNet 语言本体论中开发了一种自动概念选择算法。

图 14.7 显示了本体构建的一个例子。

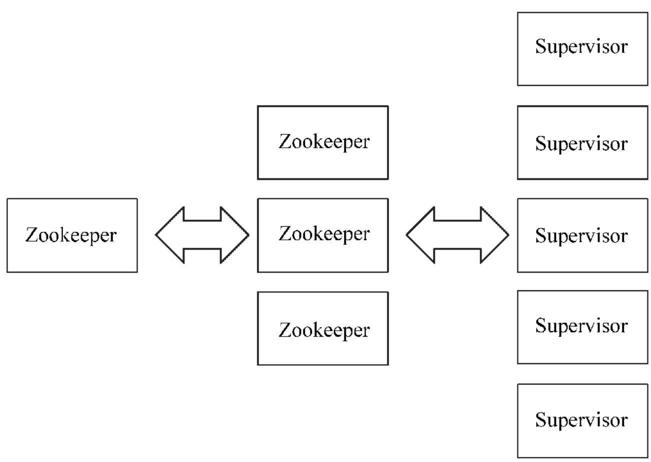


图 14.7　本体构建

14.4　走向 SNODSOC + +

我们正在研究将多种工具集成以开发 SNODSOC + +。注意，基于 SNOD 的 SNODSOC 具有一些局限性。首先，SNOD 不考虑由于流的动态性质而发生的特征演化问题（稍后将解释）。其次，如果流中出现多个新类，SNOD 就不能检测到它们。再次，SNOD 没有解决高维特征空间的问题，这可能导致更高的训练和分类误差。最后，SNOD 不对特征提取和分类应用进行任何优化。因此，我们正在开发一个实用和强大的博客和推文检测工具，我们称之为 SNODSOC。要从 SNODSOC 发展到 SNODSOC + +，

我们首先需要将 SNOD 扩展到 SNOD + + 。也就是说，我们建议将 SNOD 扩展到 SNOD + + 中，这比 SNOD 更实用和更强大。除了解决无限长度、概念漂移和概念演化问题之外，SNOD + + 还处理特征演化和多个新类，以及应用子空间聚类和其他优化，所有这些都提高了其鲁棒性、性能和精度。我们开发的所有工具（例如 LOCEXT、ONTCON 和 ENTEXT）正在与 SNODSOC 集成，来开发 SNODSOC + + 。SNODSOC + + 将基本上会实现 SNODSOC 与从推文和博客中提取的语义知识相结合。

14.4.1　SNODSOC + + 的益处

SNODSOC + + 将在社交网络分析应用程序中发挥作用，包括交互自适应检测的博客和推文。SNODSOC + + 将能够处理大量的训练数据，也将能够应付数据中的漂移。这些特性使得它比使用静态数据训练的博客和推文探测器更加实用和更加健壮。此外，它可以用于检测一个或多个新类的博客和推文。回想一下，现有的博客和推文检测技术可能无法检测出的全新模式，但是 SNODSOC + + 能够检测到这样的新类和引发警报。随后，博客将被人类专家分析和描述。特别是，SNODSOC + + 将比 SNODSOC 更健壮和更有用途，因为 SNODSOC + + 将能够在流中检测多个新的博客和推文。同时，由于其对高特征维度的鲁棒性和分布式特征抽取和选择的应用，具有较高的分类精度和更快的训练时间。

14.5　基于云的社交网络分析

如前所述，云计算作为数据密集型应用程序实现极高可扩展性的设计模型，越来越受欢迎。在云计算范例中，数据存储和检索操作是并行的，在商品硬件集群上进行。云计算解决方案已被主要生产行业领导者所采用，如谷歌、亚马逊和 Facebook。我们的目标是集成多个社交网络并分析云中的数据，如图 14.8 所示。

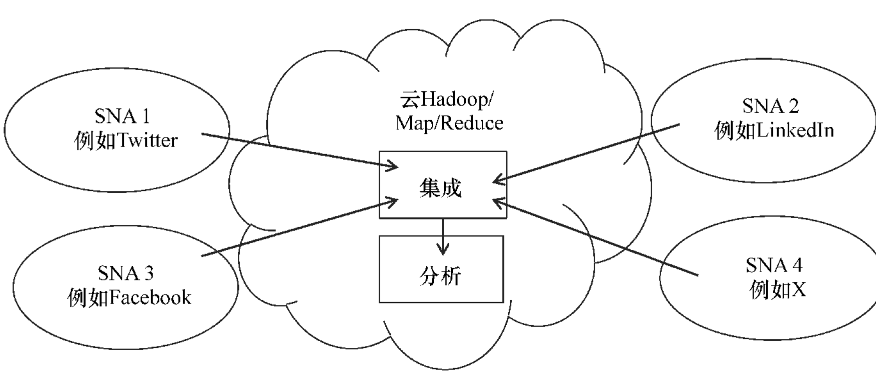


图 14.8　基于云的社交网络分析

为了开发可扩展的 SNODSOC 和 SNODSOC + + ，我们正在使用 Twitter Storm，

在 GitHub 的 Eclipse 公共许可 1.0 之下，它是一个分布式、容错、实时计算系统。Storm 是一个由 BackType 开发的实时处理系统，现在由 Twitter 进行保护授权使用。它可以被认为是 Hadoop 的实时处理系统，也就是说，它是用于 Hadoop 批量处理的实时处理。

14.5.1 流处理

Storm 可用于实时处理新数据流和更新数据库。与网络中使用队列和工作人员进行流处理的标准方法不同，Storm 具有容错性和可扩展性。Storm 集群在表面上类似于 Hadoop 集群。而在 Hadoop 上，运行“Map/Reduce 作业”，在 Storm 上运行“拓扑”。“作业”和“拓扑”本身是有很大不同的，一个关键的区别是 Map/Reduce 作业最终会完成，而拓扑会永远处理消息（或直到用户终结它）。Storm 集群上有两种节点：主节点和工作节点。主节点运行一个名为“Nimbus”的守护进程，类似于 Hadoop 的“JobTracker”。Nimbus 负责在群集周围分发代码，将任务分配给机器并监控故障情况。

每个工作节点均运行一个称为“监管者”的守护程序。监管者监听分配给其机器的工作，并根据 Nimbus 分配启动和停止它的工作进程。每个工作进程执行一个拓扑的子集，运行的拓扑结构包括分布在许多机器上的工作进程。Nimbus 与监管者之间的所有协调都是通过 Zookeeper 集群完成的。此外，Nimbus 和 Supervisor 守护进程都是快速失败和无状态的，所有状态都保存在 Zookeeper 或本地磁盘上。这意味着用户可以终结 -9 Nimbus 或监管者，他们会像没有发生任何事情一样重新开始。这种设计使得 Storm 群集相当稳定。我们已经有拓展并运行了几个月，没有进行任何维护。

14.5.2 SNODSOC 的 Twitter 风暴

数据加载模块会将 Twitter 用户数据添加到基于云的语义 Web 数据库中。对于这部分数据，将创建 Twitter 使用的术语词汇。这个词汇表是进入语义 Web 世界的最简单方法。词汇表是所有语义 Web 语言中最不具表现力的，但在处理方面却是最有效的。首先，为那些我们所希望使用的数据源（如 Google Plus、LinkedIn 等）构建相应的词汇表，然后根据这些词汇进行一定量的数据自动整合。我们的最终目标是为了对数据进行复杂的推理，后来将这些词汇细化为更复杂的本体，这些本体表现得更加丰富，但需要更长的处理时间。Twitter Storm 将为实时提取、位置挖掘和更重要的新类检测，提供一个即时处理推文和其他社交媒体消息的框架。为了查询语义 Web 数据库，需要对 14.2 节中讨论的 SPARQL 查询处理程序进行审查。目前，我们正在使用 Twitter Storm 框架来实施 SNODSOC。我们还在使用 Twitter Storm 框架开发第二个社交网络系统，称为 StormRider。StormRider 将在第 15 章讨论。

14.6 相关工作

对我们提出的方法产生影响的学者包括 Goyal、Katakis、Lin、Markou、Smith、Spinosa、Frigui、Backstrom 和 Wenerstrom 等人（见本章参考文献［GOYA09］、［KATA06］、［LIN11］、［MARK03］、［SMIT01］、［SPIN08］、［WENE06］、［BACK08］、［FRIG04］）。

存在许多工具，用于从网页和其他结构化文本中，进行实体提取和位置识别。虽然这些检测策略的一些细节是专用的，但众所周知，它们都使用标准的自然语言处理或机器学习技术，也假设文本是结构化的，并且包含具有正确语法的完整句子。另一方面，我们的工作重点是非结构化文本，它们通常是与社交媒体相关联的俚语和不完整的句子。关于基于云的流挖掘和新类检测框架，据我们所知，对于以云为中心的趋势检测工具，目前并没有显著的商业竞争。目前关于趋势检测的工作（Twitter Benjardusis 的 TwitterMonitor 和 Streaming Trend Detection 在 Twitter 中）刚刚开始，并且使用基于关键字的方法而不是选择特征向量。另外，由于我们采用了模块化的方法来创建工具，可以分别以迭代的方式对每个组件（趋势分析、实体提取、云可扩展性和本体构建的新类检测）进行改进。我们用于开发 SNODSOC ++的所有框架和工具都是开源的，并且已经在以前的研究中广泛使用，因此我们的工具将能够适应平台中的任何更改。

14.7 总结和展望

本章介绍了 SNODSOC 的设计。SNODSOC 对于必须处理数十亿个博客和消息的分析者来说将是一笔巨大的财富。例如，通过分析特定群体的行为历史，分析者将能够在不久的将来预测行为变化，并采取必要的措施。这一研究领域将刺激新的社交网络分析技术，激发新的研究领域。我们还讨论了如何使用云计算来实现 SNODSOC。

新博客每天都会充满网络空间。分析这样的博客并更新现有的分类模型是一项艰巨的任务。大多数现有的行为分析技术是手动的，需要几天才能分析单个博客示例，并提取其行为概况。即使是现有的自动化技术，也只是在一小部分训练数据样本上进行了测试。通过将我们的方法与云计算框架相结合，克服了这一障碍，并将提供高度可扩展的行为建模工具，从而在检测新模式方面实现更高的准确性。此外，现有的分析（手动或自动）技术还不能发现博客和消息的演进特征。因此，通过快速响应动态环境，我们的产品将比其他基于行为的产品具有更大的优势。

虽然 SNODSOC 比目前可用的 SNA 工具有了很大的改进，但是潜在的 SNOD 技术仍有一些局限性。例如，SNOD 缺乏检测同时出现的多个新类的能力。由于博客和消息可能具有多个并发演进，因此我们需要一个可以检测多个新类的系统。因此，我们正在扩

展 SNOD，以实现更强大的检测策略（SNOD ＋＋）来解决这些局限性。我们的目标是开发一个功能齐全、功能强大的博客分析系统 SNODSOC ＋＋。我们认为，SNODSOC ＋＋将是一个社交网络分析的创新，因为它可以处理动态数据、改变模式和新兴类的动态出现。我们将利用我们的云计算框架开发 SNODSOC ＋＋。基于 SNOD 技术的初步实验，我们认为 SNODSOC 和 SNODSIC ＋＋都将提供高度的准确性、可实时扩展和可操作性。SNODSOC ＋＋还将集成工具 LOCEXT、ENTEXT 和 ONTCON，以便在提取的语义知识中检测出高精度和不断变化的模式和趋势。最后，使用我们的云计算框架，可以开发出用于挖掘社交网络数据的可扩展性解决方案。

保护社交网络也是我们的主要目标之一。我们对社交网络的安全和隐私进行了一些研究（见本章参考文献［CARM11］）。我们的系统必须在云上实施，其实施的第一步是在云中开发有保障的信息共享功能。我们将在第七部分讨论这方面的一些实验系统。

参考文献

[ABRO09] Abrol, S., L. Khan and T. M. Al-Khateeb. MapIt: Smarter searches using location driven knowledge discovery and mining, *1st SIGSPATIAL ACM GIS 2009 International Workshop on Querying and Mining Uncertain Spatio-Temporal Data (QUeST)*, Seattle, November 2009.

[ABRO12] Abrol,S., Khan, L., Khadilkar, V., Cadenhead, T. Design and implementation of SNODSOC: Novel class detectionfor social network analysis, Proceedings of the 2012 International Conference on Intelligence and Security Informatics, p. 215--220, 2012.

[AHME09] Ahmed, M. S. and L. Khan. SISC: A text classification approach using semi supervised subspace clustering. *ICDM Workshops* 2009, pp. 1–6, Miami, FL.

[AHME10a] Ahmed, M. S., L. Khan and M. Rajeswari. Using correlation based subspace clustering for multi-label text data classification. *ICTAI*, 2, 296–303, 2010.

[AHME10b] Ahmed, M. S., L. Khan, N. C. Oza and M. Rajeswari. Multi-label ASRS dataset classification using semi supervised subspace clustering. *Proceedings of Conference on Intelligent Data Understanding (CIDU)*, Mountain View, CA, 285–299, 2010.

[ALIP10] Alipanah, N., P. Parveen, S. Menezes, L. Khan, S. Seida and B. M. Thuraisingham. Ontology-driven query expansion methods to facilitate federated queries. *Proceedings of IEEE Intl. Conference on Service-Oriented Computing and Applications (SOCA)*, Perth, Australia, 2010.

[BACK08] Backstrom, L., J. Kleinberg, R. Kumar and J. Novak. Spatial variation in search engine queries. In WWW, *Proceedings of World Wide Web Conference (WWW)*, Beijing, China, 2008.

[CARM11] Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M. and B. M. Thuraisingham. Semantic web-based social network access control. *Computers & Security*, 30(2–3): 108–115, 2011.

[CHAN11] Chandra, S. and L. Khan, Estimating twitter user location using social interactions ⊠Ö a content based approach, *The Third IEEE International Conference on Social Computing*, MIT, Boston, USA, October 9–11, 2011.

[CHEN10] Cheng, Z., J. Caverlee and K. Lee. You are where you tweet: A content-based

approach to geo-locating twitter users. *Proceeding of the 19th ACM Conference on Information and Knowledge Management (CIKM)*, Toronto, October 2010.

[FRIG04] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3): 567–581, 2004.

[GOYA09] Goyal, A., H. Daum and S. Venkatasubramanian. Streaming for large scale NLP: Language modeling. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, 2009, pp. 512–520.

[HUBE09] Huberman, B. and D. R. F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14, 2009.

[HUSAIN11] Husain, M. F., McGlothlin, J. P., Masud, M. M., Khan, L. R. and B. M. Thuraisingham. Heuristics-based query processing for large RDF graphs using cloud computing. *IEEE Transactions on Knowledge and Data Engineering*, 23(9): 1312–1327, 2011.

[FRIG04] Frigui, H. and O. Nasraoui, Unsupervised learning of prototypes and attribute weights, *Pattern Recognition*, 37(3): 567–581, 2004.

[GOYA09] Goyal, M. A., H. Daum and S. Venkatasubramanian, Streaming for large scale NLP: Language modeling, *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, 2009, pp. 512–520.

[KATA06] Katakis, I., G. Tsoumakas and I. Vlahavas. Dynamic feature space and incremental feature selection for the classification of textual data streams. *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML PKDD)*, 102–116, 2006, Berlin, Germany.

[KHAL10] Khaled, A., M. F. Husain, L. Khan, K. W. Hamlen and B. M. Thuraisingham, A token-based access control system for RDF data in the clouds. *Proceedings of IEEE Intl. Conference on Cloud Computing Technology and Science,* Indianapolis, IN, *CloudCom*, 2010, 104–111.

[KHAN02] Khan, L. and F. Luo. Ontology construction for information selection. *Proceedings of IEEE Intl. Conference on Tools with Artificial Intelligence (ICTAI)*, 122-, Washington, DC, 2002.

[LIN11] Lin, J., R. Snow and W. Morgan. Smoothing techniques for adaptive online language models: Topic tracking in tweet streams, *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, San Diego, CA, August 2011.

[LUO04] Luo, F., L. Khan, F. B. Bastani, I.-L. Yen and J. Zhou. A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles. *Bioinformatics*, 20(16): 2605–2617, 2004.

[MARK03] Markou, M. and S. Singh. Novelty detection: A review. Part 1: Statistical approaches, Part 2: Neural network based approaches. *Signal Processing*, 83, 2481–2497, 2499–2521, 2003.

[MASU10] Masud, M. M., Q. Chen, L. Khan, C. C. Aggarwal, J. Gao, J. Han and B. M. Thuraisingham. Addressing concept-evolution in concept-drifting data streams. *Proceedings of ICDM'10*, pp. 929–934.

[MASU11a] Masud, M. M., T. M. Al-Khateeb, L. Khan, C. C. Aggarwal, J. Gao, J. Han and B. M. Thuraisingham. Detecting recurring and novel classes in concept-drifting data streams. *Proceedings of Intl. Conference on Data Mining*, Vancouver, BC, Canada.

[MASU11b] Masud, M. M., J. Gao, L. Khan, J. Han and B. M. Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE TKDE*, 23(1): 859–874, 2011.

[MASU12] M. M. Masud, J. Gao, L. Khan, J. Han, K. W. Hamlen and N. C. Oza. Facing the reality of data stream classification: Coping with scarcity of labeled data. *International Journal of Knowledge and Information Systems (KAIS)*, 33(1), 213–244, Springer, 2012.

[MCGL10] McGlothlin, J. P. and L. R. Khan. Materializing and persisting inferred and uncertain knowledge in RDF datasets. *Proceedings of AAAI Conference on Artificial Intelligence*, Atlanta, GA, 2010.

[SMIT01] D. A. Smith and G. Crane. Disambiguating geographic names in a historical digital library, *5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2001), Lecture Notes in Computer Science*, Darmstadt, Germany, September 2001, pp. 127–136.

[SPIN08] Spinosa, E. J., A. P. de Leon, F. de Carvalho and J. Gama. Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. *Proceedings of ACM SAC*, 2008, pp. 976–980.

[WENE06] Wenerstrom, B. and C. Giraud-Carrier. Temporal data mining in dynamic feature spaces. *ICDM*, 2006, pp. 1141–1145.

第 15 章　基于语义 Web 的云计算实验系统

15.1　概述

在本章中，我们将论述使用语义 Web 技术的另外 3 个云计算实验系统。第一个称为 Jena – HBase 的系统，是 RDF 三元组的存储系统。名为 StormRider 的第二个系统使用 Storm 框架来托管云中的社交网络。StormRider 在其实现中还使用了 Jena – HBase。第三个系统是利用 Map/Reduce 框架的本体驱动查询处理系统。我们还描述了 3 个系统背后的动机。

第一，缺乏可扩展性是单机 RDF 数据存储所面临的最重要问题之一。云计算的出现为 RDF 三元组存储的分布式生态系统铺平了道路，这些存储可以允许与分布式查询处理功能一起实现行星级存储。为此，我们介绍了 Jena – HBase，这是一个可以与 Jena 框架一起使用的支持三元组存储的 HBase。Jena – HBase 为最终用户提供可扩展的存储和查询解决方案，支持 RDF 规范中的所有功能。

第二，在线社交媒体提供商的重点已经从“内容生成”转向寻找“内容存储、检索和分析”的有效方法，因为要适应不断发展的网络。为此，我们提出了 StormRider，该框架使用现有的云计算和语义 Web 技术，为应用程序员提供对这些任务的自动化支持，从而可以在底层不断发展的社交网络中，实现更加丰富的用例分类。

第三，鉴于需要高度分布和联合的架构，强大的查询扩展对信息检索的性能有很大的影响。我们使用不同的加权技术，来确定本体驱动的查询扩展术语。为此，我们考虑每个单独的本体和用户所查询的关键字，以使用包括中介度量（Betweenness Measure，BM）和语义相似性度量（Semantic Similarity Measure，SSM）在内的多种语义测度来确定基本扩展术语（Basic Expansion Terms，BET）。我们开发了一个 Map/Reduce 分布式算法，用于计算本体图中的所有最短路径。Map/Reduce 算法将大大提高大型本体的 BET 计算效率。

本章的结构如下：在 15.2 节讨论 Jena – HBase；StormRider 在 15.3 节中讨论；第 15.4 节讨论使用 Map/Reduce 实现的本体驱动查询处理工具；15.5 节总结本章。本章内容如图 15.1 所示。

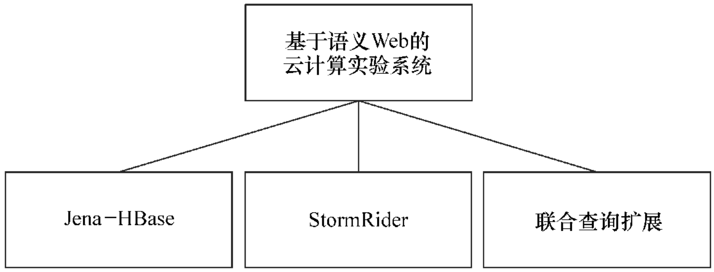


图 15. 1　基于语义 Web 的云计算实验系统

（源自：Khadilkar，V.，M. Kantarcioglu，and B. Thuraisingham，StormRider. Harnessing "Storm" for social networks. Technical report，543 – 544，2012. http：//www. utdallas. edu/vvk072000/Research/StormRider/tech – report. pdf）

15. 2　Jena – HBase：分布式、可扩展和高效的 RDF 三元组存储

存储 RDF 三元组的最简单方法是把 3 列组成关系/表，每列用于主体、谓词和客体。然而，这种方法由于缺乏可扩展性和简化了查询性能，当 RDF 三元组的数量增加时，单个表变得又长又窄（见本章参考文献［ERET09］）。该方法不可扩展，因为这些表通常位于单个机器上。云计算模式使并行实现多台机器的处理能力成为可能。Hadoop 和 HBase 等工具提供了比如容错和实时查询优化等优点。在本节中，我们介绍 Jena – HBase，一个 HBase 支持的三元组存储，可以与 Jena 框架一起使用，并对我们的原型进行初步的实验评估。

我们的工作重点是创建分布式 RDF 存储框架，从而减轻单机系统存在的扩展性问题。选择 Jena 框架的动机是它已被广泛接受，它内置支持操纵 RDF 数据以及开发本体的功能。此外，HBase 被选择用于存储层，原因有两个：①HBase 是一个面向列的存储，一般来说，一个面向列的存储性能要优于行的存储性能；②Hadoop 包括存储数据的分布式文件系统 HDFS 和用于处理存储在 HDFS 中数据框架的 Map/Reduce。HBase 使用 HDFS 进行数据存储，但不需要 Map/Reduce 来访问数据。因此，Jena – HBase 不需要实现基于 Map/Reduce 的查询引擎来执行对 RDF 三元组的查询。相反，使用基于 Map/Reduce 的查询引擎来处理 RDF 数据的现有系统能优化查询性能。然而，目前它们无法支持 RDF 规范中的所有功能。我们开发 Jena – HBase 的动机是想为最终用户提供基于云的 RDF 存储和查询 API，以支持 RDF 规范中的所有功能。

Jena – HBase 提供以下功能：①针对 HBase 中各种定制的 RDF 数据存储布局，在查询性能/存储方面提供了权衡；②通过适当的 Jena 接口的实现来支持具体化、推断和 SPARQL 处理。图 15. 2 是对 Jena – HBase 所采用架构的概述。Jena – HBase 使用存储的概念在底层 HBase 表上提供数据操作功能。一个存储代表单个 RDF 数据集，并且可以由几个 RDF 图组成，每个 RDF 图都有自己的存储布局。布局使用几个不同模式的 HBase 表来存储 RDF 三元组，每个布局在查询性能/存储方面均提供了权衡。RDF 图上

的所有操作都将隐式转换为底层布局的操作。这些操作包括：①格式化布局，即删除所有三元组同时保留表（格式化块）；②将三元组装载到布局中（装载程序块）；③查询与给定（S，P，O）模式（Query Runner block）匹配的三元组的布局；④附加操作包括：维护 HBase 连接（连接块）和维护每个 RDF 图（配置块）的配置信息。

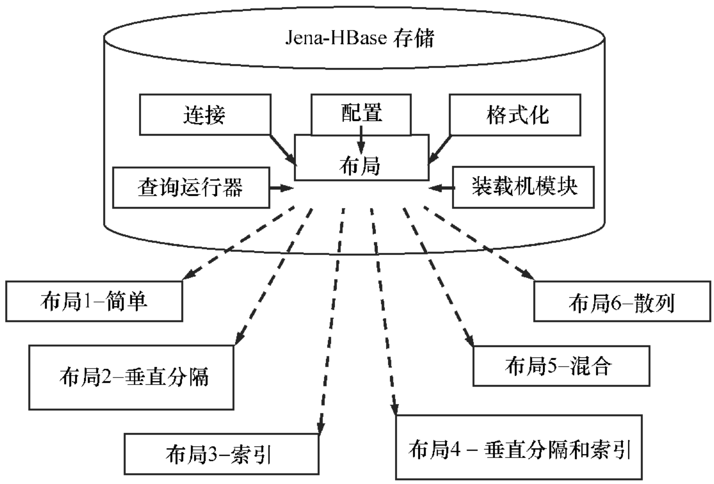


图 15.2　Jena – HBase 架构

（源自：Khadilkar，V.，M. Kantarcioglu，P. Castagna，和 B. Thuraisingham，Jena – HBase：A distributed，scalable and effcient RDF triple store. Technical report，2012.
http：//www. utdallas. edu/ ~ vvk072000/Research/Jena – HBase – Ext/tech – report. pdf）

我们已经使用 SP^2Bench（非参考查询）（见本章参考文献［POTA09］）和 LUBM（推理查询）（见本章参考文献［BRAN01］）进行基准测试，以确定 Jena – HBase 中目前可用的最佳布局，并将最佳布局的性能与 Jena TDB 进行比较。我们将 Jena – HBase 仅与 Jena TDB 进行比较，而不是与其他基于 Hadoop 的系统进行比较，原因如下：①Jena TDB 提供了所有可用的 Jena 存储子系统的最佳查询性能；②可用的基于 Hadoop 系统没有实现 RDF 规范中的所有功能。作为确定最佳布局过程的一部分，我们在几个不同图表规模上运行了两个基准测试，我们的结果在本章参考文献［KHAD12a］、［KHAD12b］中给出。由于 LUBM 包含推理查询，因此我们使用“Pellet 推理器”（v2. 3. 0）执行推理。

15.3　StormRider：利用社会网络的“Storm”

社交媒体应用程序的兴起，已经将曾经的特权领域——Web 创作和发布变成了一个常见的活动。这导致在线用户生成内容的数量爆炸式增长。社交媒体提供商的主要关注点不再是“内容生成”，而是寻找“内容存储、检索和分析”的有效方法。现在已经有了大量的研究（例如本章参考文献［ERET09］）来解决这个问题。然而，现有的工作将网络视为一系列快照，这里的快照表示给定时间段内网络的状态。因此，需要在每

个快照上单独执行不同的网络操作。实际上，在线社交网络是不断发展的实体，所以网络操作应该随着它们的发展而自动发展。此外，从这个角度来看待问题，可以让我们创建一个解决方案，支持先进的现实世界的用例，如：①跟踪给定节点的邻域，这种用例和法律实施方面是相关的。例如，跟踪潜在的罪犯/恐怖分子的活动；②能够存储和访问网络的先前快照，以进行审计和验证任务。这样的用例与医疗保健中是相关的，例如，在跟踪患者的病史方面。

在本节中，我们介绍 StormRider，该框架使用基于云和语义 Web 工具的结合，以便对不断变化的社交网络进行存储、检索和分析。此外，用户可以通过实现创建 StormRider 中提供的自定义接口，来在它们选择的网络上执行这些操作。StormRider 框架使用以下现有工具作为基本构建模块：①Storm 框架允许 StormRider 随着时间的推移和底层网络的发展而自动存储、查询和分析数据，Storm 之所以被选中，是因为它是一个实时的计算系统，保证了消息处理的可扩展性、可靠性和容错能力。②Jena – HBase 框架（见本章参考文献［KHAD12a］）允许将网络数据以一个 RDF 表示形式来存储，并使用 SPARQL 查询数据。③Apache HBase 用于构建物理视图，存储与网络中节点相关的元数据，这些视图能够在网络上执行更快的分析。

StormRider 提供以下新的贡献：①Jena – HBase 框架有助于使用如推理算法、验证等社交网络应用中的几种语义 Web 功能；②通过使用在 Storm 中实现的新算法（例如，用于中心性估计的近似算法）来存储、查询和分析，促进网络不断发展；③应用程序员提供了简单的接口，通过它们可以与他们选择的社交网络进行交互。

图 15.3 给出了 StormRider 的架构情况。用户应用程序与抽象的社交网络模型（Model – SN）进行交互，该模型将高级用户定义的网络操作（即存储、查询和分析），转换为 StormRider 所使用的低级操作，这种低级操作是基于底层网络表示的。低级操作

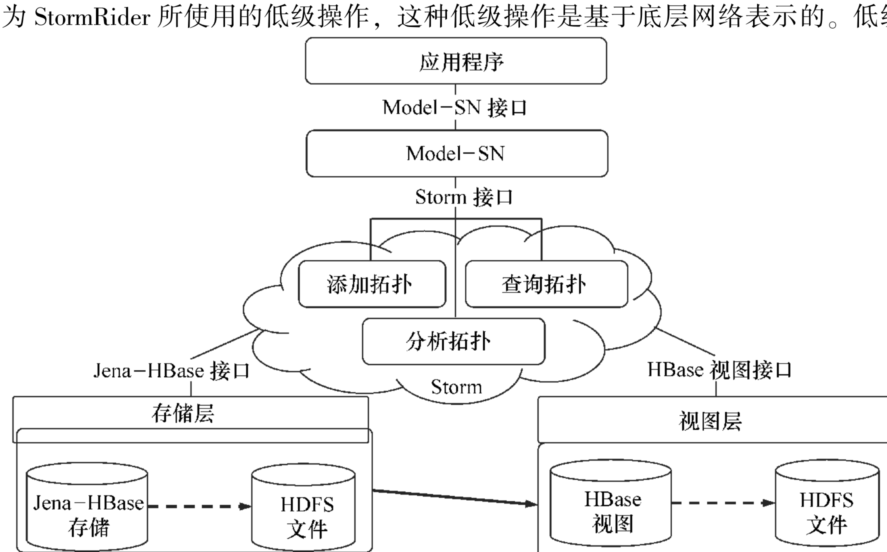


图 15.3　StormRider 架构

实施由 Storm 拓扑完成，旨在支持不断发展的社交网络。Storm 拓扑表征一个计算图，其中节点包含计算逻辑，节点之间的链接表明数据如何从一个节点传递到另一个节点。Storm 内部通过 HBase 视图接口与存储层（Jena – HBase）、Jena – HBase 接口和视图层（HBase 表用作物化视图）进行内部连接，以在底层网络上执行拓扑。

由 Jena – HBase 组成的存储层，用于在基于云的框架中的 RDF 表示中存储网络。与 Storm 中定义的拓扑结合使用时，RDF 中的网络存储支持先前给出的实际用例及其具体化工作，这中间需要使用比如属性路径查询等概念。有关 Jena – HBase 的更多详细信息，有兴趣的读者可参考我们更加详细的技术报告（见本章参考文献［KHAD12a］）。视图层用于存储组成网络中节点的元数据。元数据主要用于在分析网络期间促进性能的提升。

该架构的附加细节，以及 Twitter 网络的示例添加、查询和分析拓扑的详细描述在本章参考文献［KHAD12c］中给出。需要注意的是，这些拓扑仅作为 StormRider 框架的示例被提供。因此，应用程序员需要根据自己的要求自定义拓扑，以便与要审查的网络进行交互。

StormRider 中的示例拓扑已经在 Twitter 中实施。添加拓扑用于向存储层添加数据，以及更新视图层中节点的相关信息。然后使用分析拓扑调用视图层中的元数据来计算节点的度、亲密度和中间度。这些度量中有一些需要使用基于地标的近似技术（见本章参考文献［POTA09］）进行最短路径计算。作为我们实验评估的一部分，对最多 500000 个 Twitter 用户计算了他们的亲密度和中间度，评估了这种方法的有效性与本章参考文献［BRAN01］中给出方法的确切性。地标集中的节点数被设置为（总用户数）/ 100，其中因子 100 被随机选择，而地标集中的元素被选为具有最高度的顶端 k 个节点。最后，每个实验围绕以下方面进行：①近似误差，该度量参数用来测量 StormRider 在计算中心度值与准确方法时的准确性，并计算为 $|\hat{l}-l|$，其中 l 是实际中心值，$\hat{l}$是近似值；②执行时间，该度量参数用来测量进行中心值的近似和精确计算所需的时间。近似情况的时间计算为更新视图所需时间和执行实际中心性计算所需时间的总和。我们的实验结果在本章参考文献［KHAD12c］和［KHAD12d］中给出。

请注意，在第 14 章中，我们描述了在云端运行的社交网络 SNODSOC 的设计。SNODSOC 利用 Twitter Storm 实现云。目前，SNODSOC 和 StormRider 是在不同项目上单独开发的两个独立系统。未来，我们将探讨 SNODSOC 与 StormRider 的集成。

15.4 使用 Map/Reduce 框架进行本体驱动的查询扩展

分布式和并行计算会持续解决联合架构中许多 Web 应用程序的效率问题。由于数据应用程序在这种架构中使用分布式数据源，因此需要大量的原始用户查询，并通过查询扩展来弥补用户查询和所需信息之间的差距。许多研究人员的目标是区分不同的扩展项，提高查询扩展的鲁棒性。在我们以前的工作（见本章参考文献［ALIP10a］）中，开发了一种用于本体驱动的查询扩展的新的加权机制，称为 BET 和新的扩展项（New Expansion Terms，NET）。对于每个用户查询，基于一些度量，即语义相似性、密度和中

间性，来计算每个本体中的 BET。NET 的确定是通过对齐本体，来确定不同本体之间的健壮扩展项。使用本体图中的最短路径计算，来定义 BET 度量。通过本体图中的每个实体，找到最短路径的数量，这个问题对于大本体来说是不实际的。因此，在本节中，为了计算 BET，我们将重点介绍每个拓扑中的 Map/Reduce 算法。更多细节参见本章参考文献［ALIP10a］、［ALIP10b］、［ALIP11a］、［ALIP11b］、［ALIP12a］和［ALIP12b］。我们的查询扩展架构如图 15.4 所示。

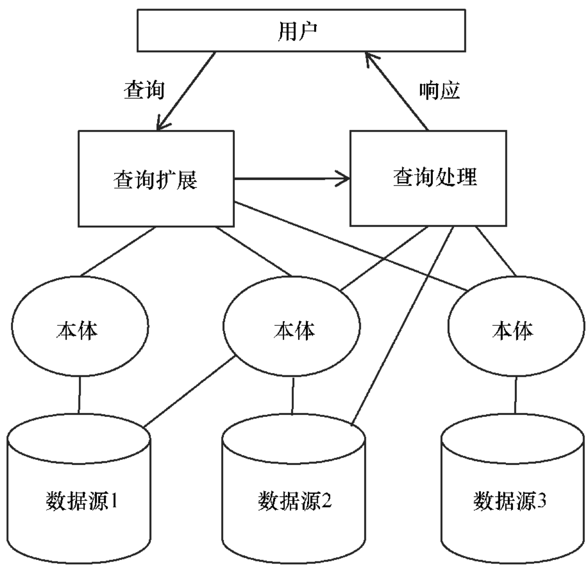


图 15.4　查询扩展架构

15.4.1　使用 Map /Reduce 分布式方法对 BET 计算

在我们的 BET 计算（见本章参考文献［ERET09］）中，我们使用 BM、中心性、密度测度（Density Measure，DM）和 SSM 度量。为此，首先我们使用每个本体中的 BM 来确定中心实体（Central Entity，CE）。接下来，我们使用 CE 并对 BET 中的每个扩展项计算 SSM。在 BM 和 SSM 度量中，我们需要按照如下方式，确定每个本体的不同实体之间的最短路径。BM 和 CE：在计算扩展项时，中间性（BM）体现为通过本体图中每个节点的最短路径数。为了扩展用户项，出现在许多最短路径上的节点被认为是每个本体（见本章参考文献［ERET09］）的中心关键字。令 e_i，$e_j \in O_k$。BM（e）是实体 e 的 BM。

$$BM(e) = \sum_{e_i \neq e_j \neq e} \left(\frac{\text{shortestpath}(e_i, e_j)\,\text{passinge}}{\text{shortestpath}(e_i, e_j)} \right) \qquad (15.1)$$

SSM 中用于查找 BET 的中心关键字由 BM 确定。中心关键字具有最高的 BM 值。SSM：SSM 使用本体图作为域的语义表示，以确定每个本体中所有扩展项的权重。靠近中心节点的实体有更多的权重。使用最短路径度量计算 SSM。实体如果有更多的共同关系，它们就越接近本体（见本章参考文献［ERET09］）。任何实体如果位于距离中心节点相对较远的

位置，则它的权重较小。因此，我们使用最短路径这个参数作为本体词汇的权重。

令实体 e_j，$c \in O_i$，并且在 c（中心）和 e_j之间存在路径。

$$\mathrm{SSM}(c, e_j) = \begin{cases} \dfrac{1}{\mathrm{length}(\mathrm{minpath})(c, e_j)} & c \neq e_j \\ 1 & c = e_j \end{cases} \qquad (15.2)$$

在这种方法中，BET 是从中心关键字到 e_j的最短路径中的所有实体。BM 和 SSM 计算中的最短路径，由宽度优先搜索（Breadth First Search，BFS）确定。给定本体图中的分支因子 b 和图形深度 d，渐近空间和时间复杂度就是最深层次的节点数，O（b^d）是指数级的。对于大的本体，最短路径计算是不实际的。因此，在下一节中，我们将介绍 Map/Reduce 分布式算法，以优化联合查询扩展。

15.4.1.1　使用迭代 Map/Reduce 算法的最短路径计算

Map/Reduce 编程模型是一个强大的接口，可以用于大规模计算的自动并行化和分配。在此模型中，Map 和 Reduce 函数定义如下：

$$\mathrm{Map}(<in_{key}, in_{Value}>) \rightarrow <out_{key}, intermediate_{Value}> list$$

$$\mathrm{Reduce}(<out_{key}, intermediate_{Value}> list) \rightarrow out_{Value} list \qquad (15.3)$$

来自数据源的数据，以数据对 $<in_{key}$，$in_{Value}>$ 的方式输入到 Map 函数中。从输入开始，经过 out_{key}，Map 函数产生一个或多个中间值。在 Map 阶段之后，任何给定 out_{key} 的所有中间值都组合成一个列表。针对同一个 out_{key}，规约功能将 $intermediate_{Value}$ 组合成一个或多个最终值。在我们的 BET 计算中，使用图 15.5 所示的本体图作为系统的输入。本体图需要从图到邻接列表的转换，以供映射函数使用。

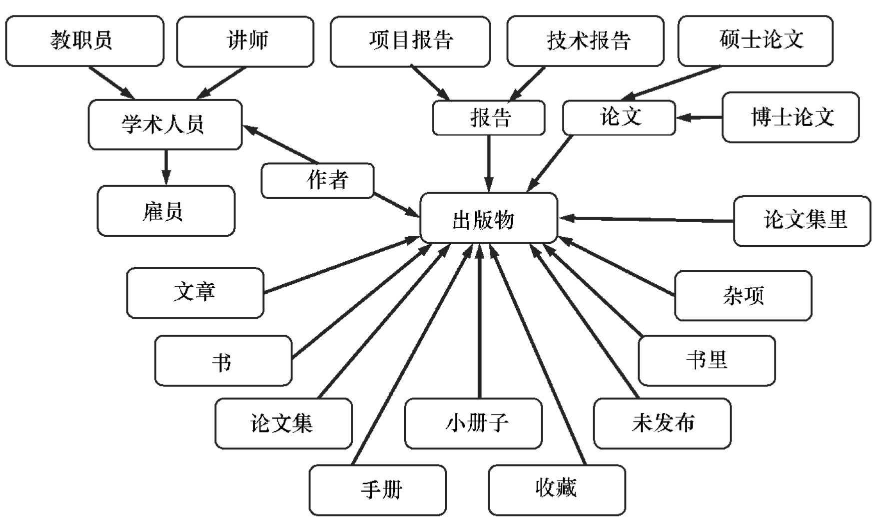


图 15.5　卡尔斯鲁厄（Karlsruhe）书目本体论

步骤 1 解释 Map 函数的转换和 $<in_{key}$，$in_{Value}>$。

步骤 1：给定本体 O_1，该算法为每个实体 $e_i \in O_1$ 构造邻接矩阵。每个 e_i 均被认为是

in_{key}。对于每个实体 e_i，我们首先确定 e_i 的邻居，并将它们存储为与该实体相连的，以逗号分隔的邻居列表（Neighbour List，NL）。我们还指定了每个实体的源距离（Distance From Source，DFS）、源路径（Path From Source，PFS）和颜色。每个实体有 3 个可能的实体颜色（Entity Color，EC）。源实体用灰色定义，而访问实体用黑色定义，而非访问实体用白色定义。对源实体，我们定义 DFS = 0，对于其他实体，定义 DFS = Int. Maxval，因为我们对源实体使用 BFS。还有，对于源实体，源的 $PFS = in_{key}$，而其他实体的 PFS 为空。NL、DFS、PFS 和 EC 的连接被认为是 in_{Value}，并作为映射函数的值。考虑图 15.5 本体图，假设 MasterThesis 是图中的源实体；因此，一些示例的键和值是

$$\langle Masterthesis,\text{"}Thesis|0|Masterthesis|Gray\text{"}\rangle,$$

$$\langle Author,\text{"}Publication,AcademinStaff|Int.Maxval|-|White\text{"}\rangle,$$

$$\langle Report\text{"}Publication|Int.Maxval|-|White\text{"}\rangle,$$

$$\langle Thesis,\text{"}Publication|Int.Maxval|-|White\text{"}\rangle$$

步骤 2：在此步骤中，映射器为 in_{key} 中的每个源实体生成 $intermediate_{Value}$。对于具有灰色的本体图中的每个源实体 Si，映射器首先将其颜色更改为黑色。然后，它基于 DFS = DFS + 1 的邻居数创建一些新的节点．那么，$PFS = in_{key} \cup in_{keynewword}$ 和 $color = Gray$。由于映射器没有关于新节点的下一个邻居的信息，所以它将下一个邻居视为空。此外，它假定非灰色节点的 PFS = “ – ”。回到我们的例子，下面是第一次迭代后的结果：

$$\langle Masterthesis,\text{"}Thesis|0|Masterthesis|Black\text{"}\rangle$$

$$\langle Thesis,\text{"}Null|1|Masterthesis\text{–}Thesis|Gray\text{"}\rangle,$$

$$\langle Author,\text{"}Publication,AcademinStaff|Int.Maxval|-|White\text{"}\rangle,$$

$$\langle Report\text{"}Publication|Int.Maxval|-|White\text{"}\rangle,$$

$$\langle Thesis,\text{"}Publication|Int.Maxval|-|White\text{"}\rangle.$$

步骤 3：在此步骤中，规约器（Reducer）使用一个 out_{key} 和所有 $intermediate_{Value}$ 的列表，来计算最终的 out_{Value}。每个 Reducer 都使用每个 *Key* 的所有 $intermediate_{Value}$，并使用边缘的“Not – Null”列表、最小 DFS、PFS 的“Not – Null”值和最暗颜色来构建一个新节点。也就是说，对于上面的例子，有

$$\langle Thesis,\text{"}Publication|1|Masterthesis\text{–}Thesis|Gray\text{""}\rangle.$$

在规约步骤 3 中的 $intermediate_{Value}$ 之后，该算法将继续进行下一个迭代，以探索第 2 步中的新的灰色节点。该迭代继续，直到它遍历了所检查的每个源的所有可能路径，并且所有节点均变为黑色。

15.4.1.2　使用 Map/Reduce 计算的中间性和中心度测度

在 BET 计算中，首先，对于每个 $q_i \in query_{keywords}$，我们计算 BM 测度。其次，我们确定所有查询关键字中的中心 $query_{keywords}$（即 q_c）。如上一节所述，在每次迭代中，中间值被更新的同时 PFS 被确定。对于每个 q_i 的中间性，在 Map/Reduce 步骤的最后一次迭代

中，我们能够发现 q_i 出现在本体中实体之间的最短路径中的次数。在确定了 BM（q_i）之后，我们指定中间性的最大值为 q_i，并作为中心关键字。

15.4.1.3 使用 Map/Reduce 算法的 SSM

我们使用上一节的 CE，并计算所有 BET 的 SSM。对于 SSM 计算，我们使用算法 1 中描述的 Map/Reduce - ShortestPath（MRSP）算法的结果。在第 2 行和第 3 行中，该算法使用 MRSP 和 SSM，使用 Map/Reduce - ShortestPath，并为每个实体返回 SSM $\in$ BET。

算法 1：使用 Map/Reduce 计算的 SSM

输入：BET $\{b_1, b_2, \cdots, b_n\}$，CE 和本体 O

输出：每个 b_i 的 SSM

1. For all $b_i \in B$ do
2. MRSP = Map/Reduce - ShortestPath result（b_i）
3. SSM（b_i） $= \dfrac{1}{\text{lengthofMRSP}}$
4. end for
5. return SSM（b_i）

15.5 总结和展望

在本章中，我们描述了 3 个各自基于云的虚拟 Web 数据管理实验系统。第一个系统将分布式 RDF 存储框架与现有的云计算工具进行集成，从而实现可扩展的数据处理解决方案。与单机 RDF 存储框架（即 Jena TDB）相比，我们的解决方案将查询执行时间开销保持在合理范围之内。接下来，我们介绍了 StormRider 框架，该框架使用现有云计算和语义 Web 技术的新颖组合，来实现对不断发展的在线社交网络的“存储、检索和分析”，从而支持几种新的、现实的使用案例。最后，我们讨论了云中的本体驱动的查询扩展系统。

还有很多领域等待进一步研究。首先，需要对我们的系统进行广泛的实验，看它们是否可以扩展。接下来，需要开发运行在云上的更强大的算法。最后，需要用一些现实世界的例子来测试我们的系统。

参考文献

[ABAD07] Abadi, D.J., A. Marcus, S. Madden, and K.J. Hollenbach, Scalable semantic web data management using vertical partitioning. In *Proceedings of Very Large Database Conference (VLDB)*, Vienna, Austria, 411–422, 2007.

[ALIP10a] Alipanah, N., P. Srivastava, P. Parveen, and B.M. Thuraisingham, Ranking ontologies using verified entities to facilitate federated queries. *Proceedings of Web Intelligence Conference*, Toronto, Canada, 332–337, 2010.

[ALIP10b] Alipanah, N., P. Parveen, L. Khan, and B. Thuraisingham, Ontology-driven query expansion methods to facilitate federated queries. *Proceedings of 2010 IEEE*

International Conference on Service Oriented Computing and Applications (SOCA10), Perth, Australia, 2010.

[ALIP11a] Alipanah, N., L. Khan, and B. Thuraisingham, Ontology-driven query expansion methods to facilitate federated queries, Technical Report, UTDCS-30-11, The University of Texas at Dallas, 2011.

[ALIP11b] Alipanah, N., P. Parveen, L. Khan, and B.M. Thuraisingham, Ontology-driven query expansion using Map/Reduce framework to facilitate federated queries. *Proceedings of International Conference on Web Services (ICWS)*, Washington, DC, 712–713, 2011.

[ALIP12a] Alipanah, N., L. Khan, and B.M. Thuraisingham, Optimized ontology-driven query expansion using Map-Reduce framework to facilitate federated queries. *Intl. Journal of Computer Systems Science and Engineering*, 27(2), 2012.

[ALIP12b] Alipanah, N., Federated query processing using Ontology structure and ranking in a service oriented environment, PhD thesis, The University of Texas at Dallas, 2012.

[BRAN01] Brandes, U., A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[ERET09] Erétéo, G., M. Buffa, F. Gandon, and O. Corby, Analysis of a real online social network using semantic web frameworks. In *Proceedings of IEEE Intl. Semantic Web Conference (ISWC)*, Chantilly, VA, 180–195, 2009.

[KHAD12a] Khadilkar, V., M. Kantarcioglu, P. Castagna, and B. Thuraisingham, Jena-HBase: A distributed, scalable and effcient RDF triple store. Technical report, 2012. http://www.utdallas.edu/~vvk072000/Research/Jena-HBase-Ext/tech-report.pdf.

[KHAD12b] Khadilkar, V., M. Kantarcioglu, B.M. Thuraisingham, and P. Castagna, Jena-HBase: A distributed, scalable and efficient RDF triple store. *International Semantic Web Conference (Posters & Demos)*, Boston, MA, 2012.

[KHAD12c] Khadilkar, V., M. Kantarcioglu, and B. Thuraisingham, StormRider: Harnessing "Storm" for social networks. Technical report, 543–544, 2012. http://www.utdallas.edu/~vvk072000/Research/StormRider/tech-report.pdf.

[KHAD12d] Khadilkar, V., M. Kantarcioglu, and B.M. Thuraisingham, StormRider: Harnessing "storm" for social networks. *Proceedings of World Wide Web Conference (WWW)*, Lyon, France, 543–544, 2012.

[POTA09] Potamias, M., F. Bonchi, C. Castillo, and A. Gionis, Fast shortest path distance estimation in large networks. In *Proceedings of Conference on Information and Knowledge Management (CIKM)*, Hong Kong, China, 867–876, 2009.

第四部分总结

第三部分讨论了云计算概念，相应地，第四部分讨论了云计算实验系统，这些是我们在 UTD 开发的系统。

在第 13 章中，我们提出了一个能够处理大量 RDF 数据的框架。由于这个框架是基于 Hadoop 的，它是一个分布式、高度容错的系统，它具有很高的可扩展性。为了增加系统的容量，需要做的所有工作就是向 Hadoop 集群添加新的节点。我们开发了一种存储 RDF 数据的模式和一种算法，来确定一个查询处理方案，以回应 SPARQL 查询。

第 14 章描述了 SNODSOC 的设计。SNODSOC 是一种基于我们开发的数据挖掘算法的社交网络系统，称为 SNOD（基于流的新颖类检测）。SNODSOC 对于必须处理数十亿个博客和消息的分析者来说，将是一个巨大的财富。例如，通过分析特定群体的行为历史，分析者将能够在不久的将来预测其行为变化，并采取必要的措施。

第 15 章描述了我们开发的 3 种基于云的工具。第一，我们提出了 StormRider，这是一个使用现有云计算和语义 Web 技术的新颖组合的框架，允许对不断发展的在线社交网络进行“存储、检索和分析”，从而支持几种新的、现实的用例。第二，我们表明，使用现有的云计算工具创建分布式 RDF 存储框架，可以获得可扩展的数据处理解决方案。此外，与单机 RDF 存储框架相比，我们的解决方案保持合理的查询执行时间开销。第三，我们讨论了云中的基于本体的查询扩展。

第五部分　云计算安全概念

第五部分简介

我们已经给出了云计算的概述，并讨论了一些云计算实验系统，现在我们将讨论云的安全问题。具体来说，我们将讨论云计算安全的概念、功能、准则和产品。

第五部分由 6 章组成：第 16、17、18、19、20 和 21 章。第 16 章将概述云计算安全概念。第 17 章讨论云计算安全的功能。云数据管理安全是第 18 章的主题。云计算安全指南在第 19 章中讨论。安全即服务的概念是在 20 章中讨论。云计算安全的产品在第 21 章中讨论。

第 16 章　云计算安全概念

16.1　概述

为许多应用程序部署云，当中遇到的主要障碍之一就是安全性。正如我们在本书前面部分所讨论的那样，通过云计算，外包给云服务提供商（Cloud Service Provider，CSP）去做的不仅是加工处理，而且数据也可以外包。因此，许多组织不愿将其数据放在云中，尤其是敏感数据。即使在云中存放加密数据，以当前技术而言，在对数据执行操作之前必须要对其进行解密。存储和管理任何形式的未加密数据，通常对许多组织而言都是不能够接受的。因此，这些组织无法利用云进行存储数据。

由于这些挑战，确保云安全已经成为许多组织迫切的需求。人们已经花费了大量资源用来确保云安全。在本章中，我们将介绍云的安全问题。我们的工作受到 10 个信息系统安全专业（Certified Information Systems Security Professional，CISSP）认证模块的影响，这些模块涉及安全性（见本章参考文献［HARR10］），同时，也受到马瑟（Mather）等人关于云安全的优秀书籍的影响。特别地，我们将回顾涉及安全概念的 10 个 CISSP 模块，并审查它们是否适合于云。这些模块如下：

- 信息系统安全和管理
- 安全架构
- 安全/访问控制模型
- 加密
- 网络安全
- 数据和应用安全
- 法律方面，包括隐私和取证
- 业务连续性规划和灾难恢复
- 物理安全
- 操作管理

本章的结构如下。16.2 节将讨论云计算的安全和管理。16.3 节将讨论云计算的安全架构。16.4 节将讨论云的访问控制和身份管理。16.5 节将讨论云的数据和应用安全问题。16.6 节讨论云的隐私、合规性和取证。密码解决方案将在 16.7 节中讨论。16.8 节将讨论云的网络安全问题。业务连续性规划将在 16.9 节进行讨论。操作安全性将在 16.10 节中讨论。物理安全问题将在 16.11 节讨论。本章总结在 16.12 节中给出。图 16.1 说明了本章讨论的各个方面。

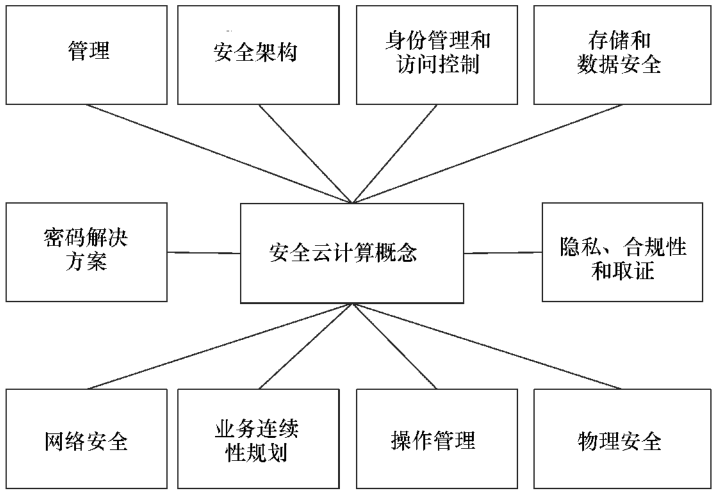


图 16.1　云计算安全的概念

16.2　云计算的安全和管理

回想一下，云框架是由 3 个主要模型组成。它们如下：

■ SaaS：软件即服务

■ PaaS：平台即服务

■ IaaS：基础设施即服务

此外，还有 3 种部署模式。它们如下：

■ 公有云

■ 私有云

■ 混合云

我们对安全问题的讨论将受到第四部分讨论中的云框架和部署模型的影响。

如本章参考文献［MATH09］所述，IT 的 5 个管理层面如下：

■ 网络

■ 存储

■ 服务器

■ 服务

■ 应用

对于内部的部署托管，组织机构可以控制存储、服务器、服务和应用程序，同时提供商和组织机构可以共享对网络的控制。例如，对于内部部署托管，组织机构通常将购买硬件、软件以及应用程序。因此，组织机构能够控制这些资源。对于网络，组织机构将与提供网络服务的提供商合作共同来完成。

对于 SaaS 模型，其所有层都由提供商控制。这是因为提供商在托管网络、平台、基础设施和应用程序。该组织机构将在提供商的基础设施和软件上，运行提供商提供的应用程序。对于 IaaS 模型，应用程序由组织机构控制，而服务由组织机构和提供商共同控制。这是由于要托管该组织机构的应用，就必须提供一些服务。然而，提供商提供如存储和网络的其余层。对于 PaaS 模型，应用程序和服务由组织机构和提供商控制，而服务器、存储和网络由提供商控制。

云的安全和管理涉及的其他方面还包括风险评估。部署云的第一步是分析其所涉及的风险。这将是由独立团队进行的详细风险分析，包括评估云的漏洞。如果要在公共环境中部署云，那么安全风险将大于在私有环境中部署云。为了适用于云，必须对 IT 系统的传统风险分析方法进行审查。

管理作为另一个方面，指的是为云制定安全策略。这将包括与这些 CSP 相对应组织的角色和责任的定义。此外，必须明确阐明和记录资产管理、人力资源安全、物质环境安全、通信和运营管理、访问控制、信息系统获取、开发和维护、信息安全事件管理、合规性和密钥管理等问题。例如，访问控制管理方面包括：谁应有访问权，为什么？资源应如何访问？如何监控访问？以及增加访问控制对 SaaS、PaaS 和 IaaS 的影响。这些方面将在随后的章节中讨论。

标准在管理 IT 系统的安全性方面也起着重要的作用。这包括安全管理标准，如信息技术基础设施图书馆（Information Technology Infrastructure Library，ITIL）和国际标准组织（International Standards Organization，ISO）27001/27002。这些标准论述了安全方面的策略、程序和进程，而且还包括可用性管理、访问控制、漏洞管理、补丁管理、配置管理事件响应和系统使用以及访问监控。

总之，为了信息系统的管理，已经开发了若干策略、进程、标准、准则和技术。为了适用于云，还应对它们进行审查和最大化的扩展。此外，还应为云计算，开发适当的风险分析技术。图 16.2 说明了管理包含的问题。

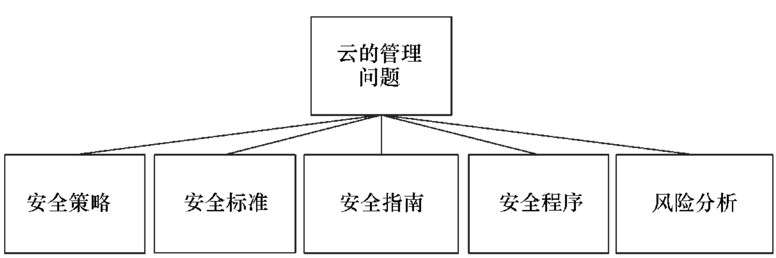


图 16.2　云的管理问题

16.3　安全体系结构

安全体系结构由系统架构的安全关键组件组成。例如，可信任计算基（Trusted

Computing Base，TCB）是一个计算系统，它由负责执行安全关键功能的部分系统组成。引用监视器是实现可信计算基系统的一部分。为了适应云，我们需要研究这些概念。具体来说，我们需要确定云关于 IaaS、PaaS 和 SaaS 的安全架构。

云的安全性将会产生不同层次的影响，这包括网络、主机和应用程序。我们将在后面的部分论述网络层次的安全问题。关于主机安全性，PaaS 和 SaaS 都将终端用户的主机操作系统隐藏起来。也就是说，SaaS 和 PaaS 中的主机安全责任被移交给 CSP。IaaS 的主机层次安全性是虚拟化的。本质上，它处理虚拟机管理程序层的安全性和客户操作系统层的安全性。

这里的主要问题：管理程序是应该直接监视客户操作系统，还是应该为每个客户操作系统安装一个监视器？如果管理程序要实时监视客户操作系统，则管理程序代码量可能会很大，这样的结果是不可取的。此外，如果管理程序发生更改，则将很难迁移到新的管理程序。这种方法的优点是管理程序可以确保所有客户操作系统的安全性。另一方面，如果客户操作系统正在进行监控，则更容易迁移到新的管理程序。这些必须在虚拟化组件层的设计中进行权衡。图 16.3 说明了监控客户操作系统的虚拟机监控程序，而图 16.4 说明了执行监控的客户操作系统。

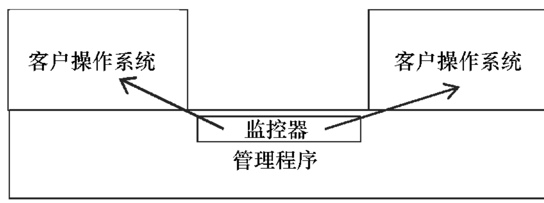


图 16.3　管理程序监控

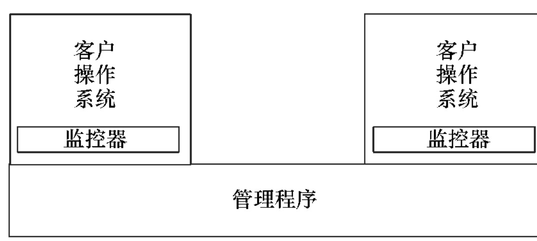


图 16.4　客户端操作系统监控

接下来，我们研究 SaaS、PaaS 和 IaaS 层次应用程序的安全性。SaaS 提供商负责提供应用程序的安全性。关于 PaaS，必须在 PaaS 平台层次以及部署在 PaaS 平台上的客户应用程序层面提供安全性。至于 IaaS，客户应用程序则被视为黑箱。也就是说，IaaS 对应用程序的安全性不负责任。

我们在本节一开始就指出安全体系结构通常由系统架构的安全关键组件组成。因此，这里的主要问题是：什么是云的 TCB？应该信任多少管理程序？安全架构师面临的挑战，是为云设计以最小化 TCB，却能提供最大化安全性的软件。

16.4　身份管理和访问控制

在本节中，我们将讨论身份管理和访问控制的技术，及其对云计算的适用性。身份管理的首要问题是信任边界。在传统的环境中，信任边界在组织机构的控制之内。这包括网络、服务器、服务和应用程序的管理。在云环境中，信任边界是动态的，并在服务

提供商以及组织机构的控制之内移动。对于在组织协作模式中处理动态和松散耦合的信任关系，身份联盟是新兴行业的最佳实践。身份联盟架构的核心是目录服务，它是身份、凭证和用户属性的存储库。身份管理使组织机构能够实现访问控制和可操作的安全性。

在本章参考文献［MATH09］中讨论了需要身份管理的各种云用例。其中包括以下内容：组织机构中的员工使用身份联盟访问 SaaS；开发人员为 PaaS 中的合作伙伴用户创建账户；终端用户访问云中的存储服务。应用程序居于一个 CSP 之中，从另一个云服务访问存储，并快速为用户提供资源以适应其变化的角色。

身份管理的 3 个主要组成部分是认证、授权和审核。认证是核实用户、系统或服务的身份。授权是给用户、系统或服务在进行身份验证后应具有的权限（例如，访问控制）。审核是检查用户、系统或服务执行的操作，并检查合规性。身份管理过程包括用户管理（用于管理身份生命周期）、身份验证管理、授权管理、访问管理、监控和审核、配置、凭证和属性管理、权限管理、合法性管理和身份联盟管理。如本章参考文献［MATH09］中所述，使用云的组织机构必须计划为用户配置账户。例如，在云中验证用户，身份管理也可以作为服务提供。

16.4.1　云身份管理

在基于云的身份管理中，必须执行用户身份的生命周期管理。我们正探索在云中实施身份联盟管理与单点登录。CSP 的责任和组织/企业的责任分别是什么？企业身份和访问管理（Identity and Access Management，IAM）要求包括为用户提供云服务账户。目前对云身份管理的研究，还包括企业如何将其身份管理需求扩展到 SaaS、PaaS 和 IaaS。

为身份管理已经制定了几项标准。身份联盟管理标准包括安全声明标记语言（Security Assertions Markup Language，SAML）、WS 联盟、自由联盟、服务配置标记语言（Service Provisioning Markup Language，SPML）、可扩展访问控制标记语言（eXtensible Access Control Markup Language，XACML）、开放授权（Open Authorization，OAuth）、开放 ID、信息卡和开放认证（Open Authentication，OAUTH）。在本节中，我们将讨论这些标准。

SAML：如本章参考文献［SAML］所述，安全声明标记语言（SAML）是一种基于 XML 的开放标准数据格式，用于在身份提供商和服务提供商之间交换身份验证和授权的数据。

WS 联盟：如本章参考文献［FEDE］所述，WS 联盟是由 IBM 和微软等不同公司开发的身份联盟规范，也是 Web 服务安全框架的一部分。它定义了允许不同安全领域代理关于身份、身份属性和身份验证的信息机制。

自由联盟：如本章参考文献［LIBE］所述，2001 年 9 月由各组织成立的自由联盟，制定了身份管理的开放标准、指南和最佳实践。

服务配置标记语言：如本章参考文献［SPML］中所述，服务配置标记语言（SPML）是一种基于 XML 的框架，用于在合作组织之间交换用户、资源和服务配置的

信息。

可扩展访问控制标记语言：如本章参考文献［XACML］所述，可扩展访问控制标记语言（XACML）是定义以 XML 实现的声明方式访问控制策略语言的标准，以及描述如何根据策略中定义的规则评估授权请求的处理模型。

开放授权：如本章参考文献［OAUT1］所述，开放授权是授权的开放标准，允许用户与其他站点分享其存储在一个站点上的私人资源（例如照片、视频），而无需提供其凭证。相反，用户通常会提供用户名和密码令牌。开放授权提供云服务 X 从云服务 Y 访问数据的能力，而不公开其凭证。

开放 ID：如本章参考文献［OPEN］所述，开放 ID 是一种开放标准，用于描述用户如何以分散的方式进行身份验证。它不需要为自己的系统提供服务，并允许用户整合其数字身份。

信息卡：如本章参考文献［INFO］所述，信息卡是个人可以在线使用的个人数字身份。它是身份元系统的主要组成部分，它是数字身份的互操作架构，能够让人们使用基于多种技术的数字身份集合。

开放认证：如本章参考文献［OAUT2］所述。开放认证是一种行业的广泛协作，通过统一采用强认证对现有的开放标准来修改实现开放参考架构。

总而言之，我们已经讨论了云身份管理的各个方面。图 16.5 说明了这些方面。

- 安全声明标记语言
- WS联盟
- 服务配置标记语言
- 可扩展访问控制标记语言
- 开放授权
- 开放ID
- 信息卡
- 开放认证
- 开放认证API

图 16.5　身份管理和访问控制标准

16.5　云存储和数据安全

在保护云数据的同时，需要识别不同类型的数据。这些包括传输中的数据和静止的数据。数据在云中从一个节点移动到另一节点。传输中的数据必须是安全的。存储在云中的数据也必须被保护。其他数据安全问题包括数据沿袭和数据来源。那就是数据来自哪里？我们可以相信这些数据吗？数据准确吗？数据残留也是云的重要方面。也就是说，一旦客户从云端移除数据，CSP 必须确保该客户的数据不会被保留在云端。

数据安全解决方案包括加密、身份管理和清理。尽管传输中的数据被加密，但在云中使用数据则需要解密。也就是说，云中有未加密的数据。只要数据未加密，就会存在重大安全隐患。因此，敏感数据不能存储在公有云中。由克雷格·金特里（Craig Gentry，见本章参考文献［GENT09］）在斯坦福大学开发的同态加密解决方案是未来云计算的解决方案。在这种方法中，作者已经证明不必解密数据就可以执行操作。

这里主要的挑战是提供商应该收集哪些数据。例如元数据，以及如何保护这些数据？其他数据的安全问题包括访问控制和加密密钥管理。机密性、完整性和可用性是云中数据安全的目标。

数据安全还包括数据管理安全。也就是说，如何保护在云上运行的数据库？我们的大部分研究都是关于开发用于查询处理的云数据安全管理器。本书将在第六部分讨论我们的原型。图 16.6 说明了云存储和数据安全性的各个方面。

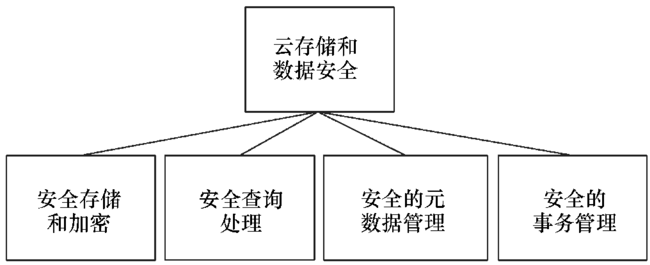


图 16.6 云存储和数据安全

16.6 云的隐私、合规性和取证

我们将法律问题分为 3 个部分。一个是隐私，另一个是审计和合规性的各种规定，第三个是取证。

16.6.1 隐私

隐私是云中的主要关注点。问题是：谁负责隐私？是数据的所有者还是云呢？数据生命周期是隐私的一个重要方面。这包括数据的生成、使用、传输、转换、存储、归档和销毁。我们需要针对数据生命周期中的每个操作制定策略。

这里需要回答几个问题，包括：谁拥有数据？是首先收集信息的组织、收集数据的人还是 CSP 呢？如果 CSP 不是数据的所有者，那么 CSP 的作用又是什么呢？需要注意的是，组织可以转移责任，但不能追究责任。评估和降低风险必须在整个数据生命周期内进行。此外，组织和 CSP 必须对法律的责任有明确认知。

许多关于数据生命周期方面的原则已经被制定出来。这些包括收集限制原则、使用限制原则、安全原则、保留和销毁原则、转移原则和责任原则。为了适应云，这些原则必须进行审查。

16.6.2 条例和合规性

为保护数据和信息以及信息技术系统，无论在美国国内还是国外，人们已经制定了几项法规。美国的隐私管理条例包括“联邦民事诉讼规则”“美国爱国者法”“电子通信隐私法案” “金融服务和市场法案（Financial Services and Markets Act，FISMA）”、“金融服务现代化法案（Gramm - Leach - Bliley Act，GLBA）”“健康保险准入和责任法案（Health Insurance Portability and Accountability Act，HIPAA）”和“健康信息技术经济与临床健康法案（Health Information Technology for Economic and Clinical Health，HITECH）”。“国际规则”包括欧盟（European Union，EU）指令和亚太经合组织（Asia

Pacifc Economic Cooperation，APEC）隐私框架。我们必须审查这些条例的云安全性。

审计和合规性是 IT 安全的主要方面。步骤包括：制定目标，定义需求（为客户提供服务），定义架构（即，架构师和结构服务以满足需求），定义策略，定义流程和程序，持续运营，持续监控和持续改进。审计规则包括“萨班斯 – 奥克斯利法案”“支付卡行业数据安全标准（Payment Card Industry Data Security Standard，PCI DSS）”、“HIPAA”和“信息及相关技术控制目标（Control Objectives for Information and Related Technology，COBIT）”。对于云计算而言，需要回答这几个问题，其包括：云计算对上述规定的影响是什么？云的内部和外部审计是什么？什么是适合云的审计框架？

16.6.3 云取证

与云相关法律方面的第三个组成部分是进行云取证。这里有两个问题。一个是使用云来进行取证。这是因为取证数据的分析可能非常耗时。因此，使用云，可以将获得取证作为服务。另一方面是分析被攻击的云。这是一个重大挑战，因为攻击可能发生在云中的任何地方。检查人员应该找出已经被破坏的节点，并确定攻击发生的时间、地点和原因。同样利用虚拟化技术，管理程序和客户操作系统一样必须被监视。有关安全架构部分的内容已经讨论了一些挑战性问题。图 16.7 说明了云的各种法律方面内容。

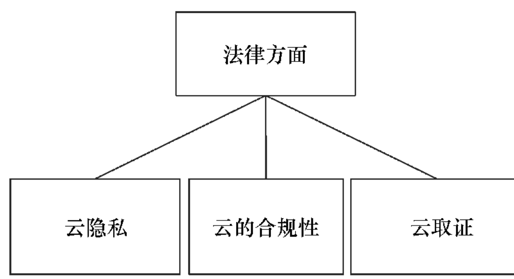


图 16.7　云的法律方面内容

16.7 加密方案

我们讨论加密是我们对云数据安全和存储概述的一部分。如前所述，所有敏感数据，无论是否在传输，都必须在云中加密。这里主要的挑战是执行操作时数据的解密。无论何时哪怕是临时的，解密的数据都必须存储在云中，这是一个重大的漏洞。虽然同态加密是一个有希望的解决方案，但这个解决方案的实现将是很多年以后的事了。

目前，存储在云中或通过云传输的数据使用传统的加密技术。这些包括对称密钥或基于非对称密钥的加密算法。例如，为了确保机密性，发送者可以使用接收者的公钥来加密数据。接收者用他的私钥解密数据。为了确保完整性，发送方可以使用他的私钥加密数据。接收方将使用发件人的公钥对数据进行解密。我们需要研究去开发适应云的特定的加密解决方案。图 16.8 为加密解决方案。

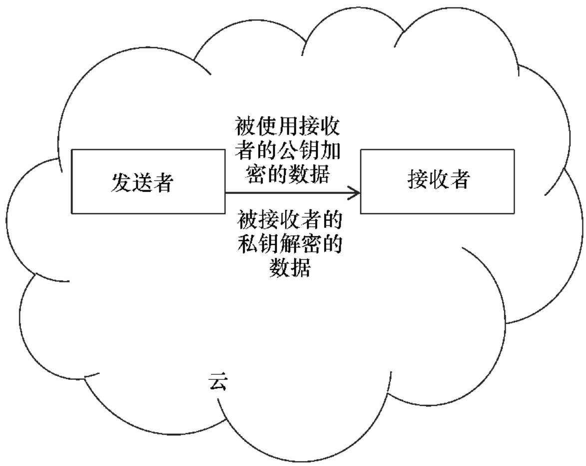


图 16.8　加密解决方案

16.8　网络安全

网络安全解决方案必须确保组织机构在公有云提供商中往返传输数据的机密性和完整性。这包括确保公有云资源的正确访问控制（身份验证、授权和审核），确保组织机构使用的公有云，在面向互联网的资源时具有可用性，以及使用域来替换已建立的网络区域和层级。也就是说，云的网络被划分为域，并且必须在域内以及跨域的情况下提供安全性。这里的一个关键是减轻风险系数。

云的大部分安全性集中在架构、数据和存储安全以及身份管理上。一些现有的网络安全协议正在应用于保护云网络。在开发云的特殊网络安全协议方面，我们还需要进行研究。图 16.9 说明了云的网络安全方面。

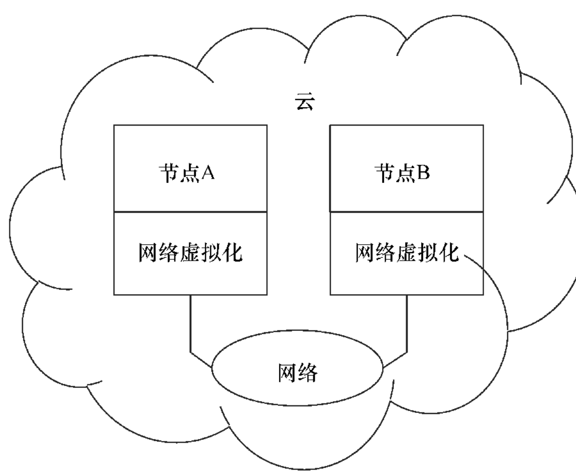


图 16.9　网络安全

16.9　业务连续性规划

IT 组织面临的挑战是业务连续性规划和灾难恢复。灾难可能包括自然灾害，如地震、飓风和龙卷风；还有人造的，如轰炸和恐怖主义。在 9・11 事件和卡特里娜之后，

许多 IT 机构发现维持运作的连续性是非常困难的。因此，适当的规划至关重要。

业务连续性规划和灾难恢复中的步骤，包括具有复制 IT 环境的镜像站点。这些站点可以是具有极少设备的冷站点或中等热度/高热度站点，并且可以被完全复制。建立备份站点并不简单。该组织必须进行模拟，以确定将可能出现的潜在问题。此外，组织必须有适当的灾难恢复计划。

在云环境中，一个主要的方面是在 SLA 中记录所有事情。例如，CSP 可能会指定云将每周停机一定的时间来进行维护，这必须记录在 SLA 中。最后，业务连续性计划和灾难管理步骤必须扩展，把云包括进去。也就是说，为了备份整个云是应该被复制，还是应该复制云的某些部分呢？我们在这方面没有太多的研究。图 16.10 说明了业务连续性规划的各个方面。

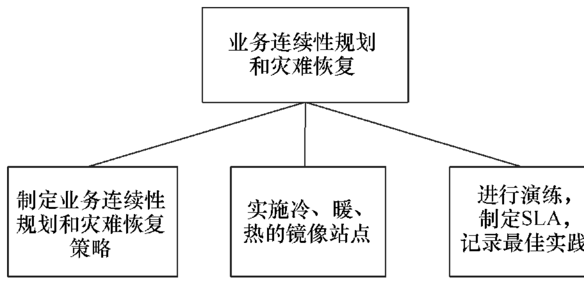


图 16.10　业务连续性规划

16.10　操作管理

IT 组织的操作管理，包括管理和维护组织中众多的计算机和网络。例如，所有的机器必须保持最新的补丁版本。这项活动通常由该组织的首席安全官管理。系统和数据库必须定期备份。此外，遗留系统必须不时地迁移到现代平台。

云中的操作管理是一个相对未开发的领域。IT 组织的操作管理将实践如何移植到云端？谁负责云中的操作管理？是否有云安全官？服务提供商的作用是什么？什么时候且如何备份云中的系统？操作管理对 SaaS、PaaS 和 IaaS 以及部署模式有什么影响？我们需要更多的研究来给出上述问题的答案。图 16.11 说明了操作管理的各个方面。

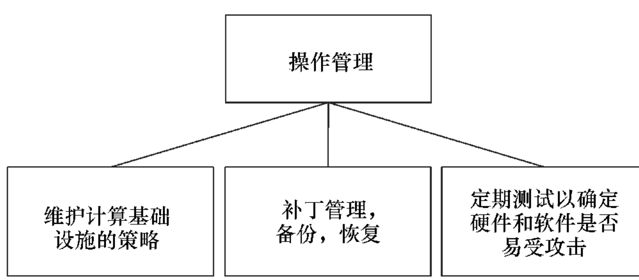


图 16.11　操作管理

16.11　物理安全

最后一点，云的物理安全性是至关重要的。对于 IT 组织的物理安全将包括保护系统和数据免受自然灾害，如火灾、人为灾害、恐怖主义和故意破坏。物理安全措施包括适当的加锁和照明，以及为栅栏、墙壁、窗户和门使用适当的材料。员工应得到允许才

能进入设备所在的楼宇。此外，除了安保人员监控进入外，还需要安装安全摄像头。

这里的问题是如何将物理安全措施扩展到云环境？云组件可能会分散在世界各地。那么谁应该保护云呢？服务提供商又有什么责任？如果云的组件被盗，会发生什么？我们需要做更多的工作来确定上述问题的答案。图 16.12 说明了物理安全性的各个方面。

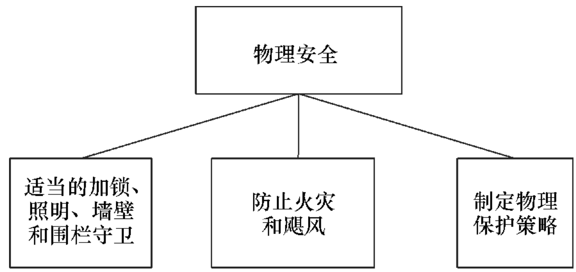


图 16.12　物理安全

16.12　总结和展望

本章讨论了各种云计算安全的概念。首先，我们讨论了云计算安全和管理。其次，讨论了云安全架构的各个方面。再次，又讨论了云的访问控制和身份管理。接着，我们讨论云的数据和存储安全问题。另外，我们概述了云的隐私、合规性和取证。最后，讨论了如加密解决方案、网络安全问题、业务连续性、操作安全性和云的物理安全问题等多个方面。

本章讨论的许多概念对于云计算来说尚未成熟。例如，关于身份管理，我们讨论了为 IT 系统中的身份管理而开发的几个标准，包括 SAML、SPML 和 XACML。这些标准必须对云适用。应用云的标准和技术的一个主要挑战是可扩展性。因为，云可以有数百个节点，并且也可以进行大量的计算，并确保客户能及时得到响应。也就是说，云的性能是一个重大问题。如前所述，云计算安全的研究才刚刚开始。在部署安全的云之前，还需要做更多的工作。

参 考 文 献

[FEDE] WS-Federation, http://en.wikipedia.org/wiki/WS-Federation

[GENT09] Craig Gentry, Fully homomorphic encryption using ideal lattices: http://www.informatik.uni-trier.de/~LEY/db/conf/stoc/stoc2009.html. STOC 2009: 169-178.

[HARR10] Harris, S., *CISSP Exam Guide*, McGraw-Hill, New York, NY, 2010.

[INFO] Information Card, http://en.wikipedia.org/wiki/Information_Card

[LIBE] The Liberty Alliance, http://en.wikipedia.org/wiki/Liberty_Alliance

[MATH09] Mather, T., Kumaraswamy, S., and Latif, S., *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance,* O'Reilly, Sebastopol, CA, 2009.

[OAUT1] Open Authorization, http://en.wikipedia.org/wiki/OAuth

[OAUT2] Open Authentication, http://www.openauthentication.org/

[OPEN] OpenID, http://en.wikipedia.org/wiki/OpenID

[SAML] Security Assertion Markup Language, http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language

[SPML] Services Provisioning Markup Language, http://en.wikipedia.org/wiki/Service_Provisioning_Markup_Language

[XACML] eXtensible Access Control Markup Language, http://en.wikipedia.org/wiki/XACML

第 17 章　云计算功能的安全

17.1　概述

在第 8 章中，我们概述了有关云的各种概念。这包括对部署模型和服务模型的讨论。在第 9 章中，我们讨论了云计算的功能。特别是，我们描述了一个云计算框架，然后又讨论了框架的各个层次。在第 16 章中，我们讨论了云计算安全的概念。特别是，我们介绍了 CISSP 模块，并探讨了云对这些模块的影响。在本章中，我们将继续讨论云计算功能的安全。具体来说，我们将研究第 9 章讨论的云计算功能，并介绍安全性对这些功能的影响。这些功能包括：

■ 云操作系统

■ 云存储系统

■ 云数据库系统

■ 云网络系统

使用我们的云计算安全框架将能更好地说明云计算功能的安全。此外，我们还将讨论云中的完整性管理。最后，我们将讨论云应用，如云信息和知识管理。

本章的结构如下。我们的云计算安全框架将在 17.2 节中讨论。17.3 节讨论操作系统功能的安全，包括虚拟化安全。17.4 节将讨论云网络安全。17.5 节将讨论云存储管理安全功能。17.6 节将讨论云数据管理功能的安全。17.7 节将讨论访问控制和身份管理等其他方面。应用功能将在 17.8 节中讨论。本章总结在 17.9 节。本章讨论的概念如图 17.1 所示。有关云计算安全问题的更多详细信息，请参见本章参考文献［HAML10］和［MATH09］。

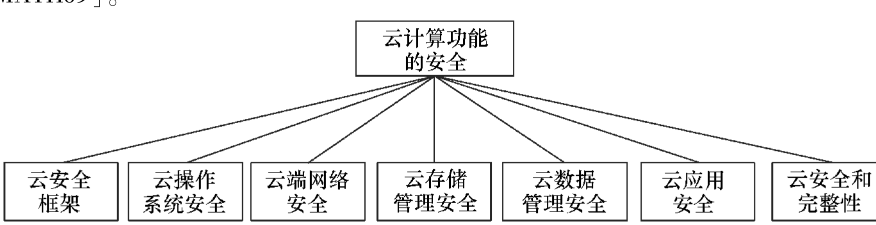


图 17.1　云计算功能的安全

17.2　云计算安全框架

云计算安全框架如图 17.2 所示。我们定义了一个分层框架，最底层是网络层，最

高层是应用层。应用程序可以是任何类型的应用程序，包括医疗保健、金融、防御和情报。我们在云端托管的应用程序包括威胁分析、恶意软件检测，以及可确保安全的信息共享和本体管理。我们将在后面的章节中描述这些应用。



图 17.2　云计算安全框架

云安全框架的核心层是操作系统/虚拟化安全层、存储安全层，以及包含数据挖掘功能的数据管理安全层。操作系统/虚拟化安全层是执行虚拟化以及内存管理、调度和进程间通信管理的层。存储安全层将管理云中数据的大量存储，其功能包括数据的加密和解密。我们已经探讨了确保 Hadoop 安全地管理分布式存储的方面。如前所述，Hadoop 与谷歌的 Map/Reduce 携手开展了分析任务。我们目前的研究是正在探索如何确保 Map/Reduce 的功能。数据管理安全层将执行云查询处理安全、云交易管理安全、云元数据管理安全和云数据挖掘安全。

第 9 章讨论了安全管理是策略管理中的一个方面。身份管理包括识别、认证和授权，其中一些概念在第 16 章中讨论过。完整性管理将包括维持数据的准确性和质量。其他与安全相关的功能包括管理、风险管理以及备份和恢复。由于在第 16 章中讨论了许多云计算安全的概念，所以不会在这里再次讨论。在本章的其余部分中，我们将讨论在本节中提到的每个功能，包括操作系统安全和虚拟机管理程序、数据存储安全、数据管理安全、网络安全、完整性管理以及应用程序安全。

17.3　云操作系统和管理程序安全

保护云操作系统涉及两个方面。一个是保护主机操作系统，另一个是保护虚拟机（Virtual Machine，VM）管理程序。保护主机操作系统包括执行适当的安全策略，以及保护如内存管理、进程间通信和调度等功能。例如，主机操作系统可以基于访问控制列表或能力表，来强制实施访问控制策略。在多层次安全的情况下，主机操作系统可能会执行如 Bell 和 LaPadula 之类的策略。进程之间的相互通信由策略决定。例如，进程 P1 可以发送消息到进程 P2 吗？访问控制策略将确定进程对文件的访问。与通常的操作系统相比，数据库系统中的访问是文件级的，而且提供了更精细的访问粒度。调度任务将涉及执行优先级较高的任务。然而，如果具有较高优先级的任务具有更高的安全级别，具有较低优先级的任务具有较低的安全级别，则可能必须中止低优先级任务，以便将资源提供给较高优先级的进程。这可能导致隐蔽的通道。在调度任务时，必须检查这些安全问题。内存管理涉及为进程运行分配内存空间。这一行动必须在不违反安全策略的情况下进行。

我们在第 9 章中讨论了虚拟化。保护管理程序是一个重大挑战。这是因为，它是使客户机操作系统分离的管理程序。也就是说，管理程序必须确保以操作系统相互不干扰

的方式分配资源。此外，客户操作系统不能破坏另一个客户操作系统的资源。因此，如果管理程序受到威胁，那么整个云可能会受到影响。因此，我们希望管理程序代码尽可能小。

这里的主要问题是，谁应该监视客户机操作系统？如果管理程序要监视所有的客户机操作系统，那么客户机操作系统就不能轻易地移植到新的虚拟机（VM）管理程序中。此外，管理程序代码可能很复杂。然而，所有的监控都驻留在虚拟机管理程序中，这意味着每个虚拟机都不必进行昂贵的计算。执行监视的管理程序称为虚拟机自审（Virtual Machine Introspection，VMI）。如果客户机操作系统进行自我监视，则管理程序代码将不会变得那么复杂，但是每个虚拟机必须执行自我监视，这可能在计算上就是复杂的。此外，VM 中的虚拟代理必须由主机操作系统进行管理，这意味着大部分安全性是主机操作系统负责。

在克里斯·本顿（Chris Benton）的文章中，他已经很好地解释了这些概念。他说，最大的安全问题是内核级的 rootkit（见本章参考文献［BENT］）。这是因为内核级的 rootkit 会将核心操作系统变成恶意软件。出现这样的结果，是因为 rootkit 具有最高级别的系统权限，并且可以利用这些权限来使自己免于检测。通过自审，虚拟机的运行权限低于虚拟机监控程序。这意味着，如果一个 rootkit 感染一个虚拟机，我们检测 rootkit 存在的能力得到了提高。

Chris Benton 说，运行防病毒软件在计算上是密集型的。因此，如果多个虚拟机同时启动完整的磁盘扫描，则 IaaS 云可能变得无法响应。他称这种情况是“反病毒（Anti－Virus，AV）风暴”。因此，在管理程序级别具备反病毒实例程序，就可以监控整个 IaaS。虽然我们期望在许多方面管理程序能够提供安全性，但也可能因为执行更多功能，而发生对管理程序的攻击。此外，虚拟机管理程序可以访问整个云，因为它可以访问所有的虚拟机。这也是一个问题。

在云中是否要执行管理程序代码，还是增强主机操作系统的操作性，这种争议仍在继续。现在的操作系统如 Android，已经支持在云中运行。最近有一些研究工作是关于云计算攻击以及攻击的解决方案。例如，北卡罗来纳大学的迈克尔·赖特（Michal Reiter）和他的团队正在研究 XEN 等虚拟机管理程序的边界攻击。得克萨斯大学达拉斯分校的林志强及其同事正在开发 VMI 解决方案，包括 VM Space Traveler。详细信息请参见本章参考文献［ZHAN12］和［FU12］。

17.4 云网络安全

在第 9 章中，我们讨论了云的网络虚拟化问题。即使在虚拟机管理程序是安全的情况下，虚拟网络的安全性仍然是一个挑战。网络虚拟化软件是否应该同时监控所有虚拟网络，还是每个虚拟网络应该有属于自己的监视器？如果单独监视，那么这些监视器要托管在物理网络上。云网络安全的另一个挑战，是将安全网络协议扩展到云中。网络协议包括 TCP/IP。这里的问题是需要将这些协议怎么扩展才能在云中运行？

Chris Benton 讨论了云中防火墙管理方面的挑战。在常规网络环境中，通常有组织良好的防火墙策略。然而，在云中，可能会有多个提供商提供多个防火墙。管理所有防火墙的策略可能会成为一场噩梦。因为这些策略不一致，并有不同的表示。其中的挑战是找出不同策略之间的一些相对统一的映射，这就是虚拟防火墙。

马瑟（Mather）、库马拉斯瓦米（Kumaraswamy）和拉蒂夫（Latif）认为，传统的网络区域和层级，必须被云环境中的安全域或安全工作组所取代（见本章参考文献［MATH09］）。他们声明，亚马逊 AWS 的安全工作组功能使虚拟机能够通过虚拟防火墙进行通信，该虚拟防火墙基于 IP 地址、端口和防火墙来过滤网络通信数据包。

总而言之，安全挑战包括保护虚拟网络、定义域名概念、通过虚拟网络和防火墙确定虚拟机之间的通信，以及审查网络协议以确定云计算的影响。

17.5 云存储管理安全

关于云存储安全，面临的挑战是为云提供适当的存储方式，而不会违反安全策略。例如，数据应该如何被分片，从而能够在云中跨节点存储，并且不会泄露任何信息？如何利用数据虚拟化来提供最佳的存储方案并保持安全性？Kantarcioglu 和 Mehrotra 已经为混合云开发了引人关注的安全存储方案（见本章参考文献［OKTA12］）。他们的存储方案考虑到对驻留在私有云中的敏感数据，以及驻留在公有云中未分类数据的查询执行成本。

加密是安全存储的另一个主要挑战。所有敏感数据如患者数据和财务数据都必须加密，且必须操纵加密的数据。虽然斯坦福大学的工作已经揭示了，如何在计算上操纵加密数据（即同态加密），但还没有开发出可行的解决方案。因此，目前敏感数据必须存储在私有云中。

我们也在研究审查 Hadoop 和 Map/Reduce 的安全性。研究如何在 Hadoop 框架上实施如 XACML 的策略模型。目前，我们的 XACML 策略是在数据级别上执行的，而不是 Hadoop 级别。我们正在探索 Hadoop 级别的模型实现。此外，我们最近的工作也包括审查 Map/Reduce 框架的安全性。

17.6 云数据管理安全

云数据管理安全问题包括云查询处理安全和云交易处理安全。我们在云数据管理安全方面做了大量工作。在后面的章节中，将讨论我们为云查询处理安全开发的原型。我们所有的原型都使用 Hadoop/Map/Reduce 框架进行分布式存储。在我们的第一个原型中，在 Hive 框架之上构建了一个 XACML 策略引擎。这里，我们假设数据位于云中的关系数据库中。在第二个原型中，我们开发了一个在云中运行的 SPARQL 查询优化器，来存储和管理 RDF 数据。查询是用 SPARQL 提出的，用 XACML 指定策略。我们已经制定了对 RDF 数据执行 XACML 策略的查询重写方案。更多详细信息，请参见本章参考文献

［THUR10］和［HUSA11］。

第三个原型，是在我们的 SPARQL 查询优化器之上，实现了基于 RDF 的策略引擎。这样就可以在数据和策略之间实现无缝集成，两者都在 RDF 中表达。策略引擎将强制执行信息共享策略，并处理推理等问题。当我们在本书的第七部分讨论云中保密信息共享时，将讨论这个策略引擎。

我们还开发了混合云的原型，利用 HBASE 等工具进行查询处理。在这里，我们假设敏感数据存放在私有云中，而未分类的数据则存放在公有云中。在这样的环境中，我们提出了用于查询处理安全的算法。详细信息可以在本章参考文献［OKTA12］中找到。

其他云数据管理安全功能包括事务处理安全、云中的数据互操作性和云数据挖掘。当我们讨论云数据管理安全时，将在第 18 章中给出关于这些功能的更多讨论。

17.7 云安全和完整性管理

云安全包括机密性、隐私和信任。现在有几个标准已经被开发出来，以提供云的安全性。我们在第 16 章讨论了这些标准。例如，SAML 等标准正在进行探索认证中。在云中运行 SAML 的可扩展性是一个挑战。为了授权，我们正在对 XACML 等标准进行研究。也就是说，一旦用户被认证，下一步就是确定用户可以访问的资源。向用户提供云资源是由 SPML 等标准来实现的。我们在第 16 章中讨论了这些标准，以及开放 ID 等其他标准。

为了确保数据和处理的完整性，必须执行各种完整性策略。全职雇员至少工作 40 小时，这样的执行策略在数据库中已经进行了广泛研究。我们还必须为云环境研究这些具有完整性的执行技术。此外，数据的来源变得至关重要，因为数据可能来自多个云中的多个应用程序。因此，产生、收集的来源数据量可能会很大。必须制定适当的策略来处理云中的大量数据。其他完整性管理技术包括容错计算、备份和恢复。这些技术的相关方面在第 16 章中讨论过，我们将在第 18 章中再次讨论。

17.8 云应用安全

在第 12 章中，我们讨论了一些云应用，如医疗保健、金融和社交网络。有关确保这些应用程序安全的更多细节，将在第 18 章中讨论。例如，考虑一个组织的知识管理。安全知识管理涉及控制组织知识产权和资源的获取，这些通常作为 Web 服务的集合被提供。例如，组织机构可能希望找到一个特定项目的专家。这可能涉及查找专家器这一 Web 服务，该服务可以通过企业内部网找到专家。这些 Web 服务可以在云上实现，以提高性能。也就是说，知识管理服务可以由 SaaS 来实现。Web 服务的安全对于确保知识管理服务至关重要。

虽然我们已经定义的框架（即操作系统和数据库系统）可在文件和数据（例如关

系表）级别提供安全性，但是应用程序应执行应用程序专用的策略。这些策略对于应用程序是唯一的。例如，在基于云的工作流的应用程序情况下，这些策略将指定一个人必须执行特定活动的授权。这些策略在工作流管理（即应用程序）级别中执行。

17.9 总结和展望

本章讨论了云计算功能的各个方面。我们概述了云计算安全框架，然后讨论了云的虚拟化、网络、数据存储和管理的安全性。实际上，我们研究了本书第三部分讨论的云的功能，并描述了这些功能的安全性。我们还有很多工作有待完成。首先，我们需要审查框架的每一层，并确定可能发生的攻击。正如我们所说，Reiter 和他的团队正在研究虚拟化层面的侧面挑战攻击。这些攻击不仅可能发生在虚拟化层面，还可能发生在存储层和数据层。我们需要对潜在的攻击和这些攻击的解决方案进行全面的研究。

在第 18 章中，我们将详细说明云数据管理功能的安全。云产品安全将在第 21 章中讨论。在本书的第六部分，我们还将描述为云数据管理安全开发的一些原型。

参考文献

[BENT] Benton, C., *The Basics of Virtualization Security, Cloud Security Alliance*, https://cloudsecurityalliance.org/wp-content/uploads/2011/11/virtualization-security.pdf

[HAML10] Hamlen, K. W., M. Kantarcioglu, L. Khan and B. M. Thuraisingham, Security issues for cloud computing. *International Journal of Information Security and Privacy*, 4(2): 36–48, 2010.

[HUSA11] Husain, M. F., J. P. McGlothlin, M. M. Masud, L. R. Khan, B. M. Thuraisingham, Heuristics-based query processing for large RDF graphs using cloud computing. *IEEE Trans. Knowl. Data Eng.* (TKDE) 23(9): 1312–1327, 2011.

[FU12] Fu, Y. and Z. Lin,Space traveling across VM: Automatically bridging the semantic gap in virtual machine introspection via online kernel data redirection. *IEEE Symposium on Security and Privacy*, San Francisco, CA, 2012: 586–600.

[MATH09] Mather, T., et al., *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, O'Reilly, Sebastopol, CA, 2009.

[OKTA12] Oktay, K. Y., V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra and B. M. Thuraisingham, *Risk-Aware Workload Distribution in Hybrid Clouds,* IEEE CLOUD, Honolulu, HI, 2012, pp. 229–236.

[THUR10] Thuraisingham, B. M., V. Khadilkar, A. Gupta, M. Kantarcioglu and L. Khan, Secure data storage and retrieval in the cloud. Collaborate Com 2010: 1–8.

[ZHAN12] Zhang, Y., A. Juels, M. K. Reiter, T. Ristenpart, Cross-VM side channels and their use to extract private keys. *ACM Conference on Computer and Communications Security*, Raleigh, NC, 2012: 305–316.

第 18 章　云数据管理安全

18.1　概述

在前两章中，我们讨论了云计算安全概念以及云计算功能安全。例如，在第 16 章中，我们讨论了云对 10 个 CISSP 主题的影响，包括管理和风险管理、访问控制、安全架构、加密、数据安全和应用程序安全、网络安全、物理安全、法律问题和取证、业务连续性管理和操作管理。在第 17 章中，我们讨论了云计算功能安全，包括虚拟化安全、云数据和存储管理安全、云网络安全和云应用安全。本章将讨论云数据管理安全。我们还将介绍包括云信息管理安全和云知识管理安全的概念。

在我们的术语中，数据由数据管理器管理。信息是从数据中提取出来的，知识就是理解信息以便进一步采取行动。数据管理技术包括数据库管理和数据管理。信息管理技术包括多媒体信息管理和协同信息管理。知识管理就是对组织的知识和专长进行再利用，来提高利润和其他收益。在本章中，我们将研究数据、信息和知识管理的安全问题，然后讨论如何将云计算技术应用于管理数据、信息和知识。我们在云查询处理安全中实现的一些原型，以及基于云的信息共享安全将在后面的章节中讨论。

本章的结构如下。在 18.2 节中，我们讨论数据管理安全，其中还包括信息和知识管理安全的讨论。在 18.3 节中，我们将讨论如何将云计算用于数据、信息和知识管理安全。本章总结在 18.4 节。图 18.1 说明了本章讨论的概念。有关数据管理安全的更多详细信息，请参见本书附录 C。

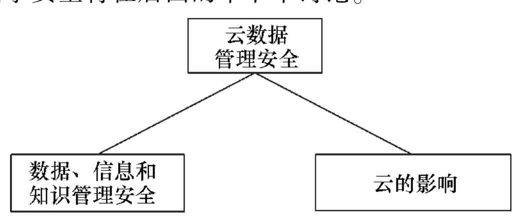


图 18.1　云数据管理安全

18.2　数据管理安全

18.2.1　访问控制

访问控制要处理对数据的授权访问，访问权限依赖用户、用户组和其他因素（如用户角色）。基于访问控制的安全性研究，最初是针对安全操作系统进行的，其中根据进程的种类授予对文件的访问权限。访问类型包括读、写操作。然后，这个概念被扩展到数据库，数据库授予对关系、属性和元素的访问权限。现在，基于访问控制的安全是

自主安全的一部分，还包括处理复杂的安全策略、基于角色和功能授予对数据的访问，以及正负授权策略。在数据管理系统中执行安全策略的技术之一是查询修改，也称为查询重写。在这种方法中，根据策略修改查询，执行修改后的查询。图 18.2 为各种类型的自主访问控制机制。查询修改如图 18.3 所示。更多细节也在本章参考文献［THUR05］中给出。

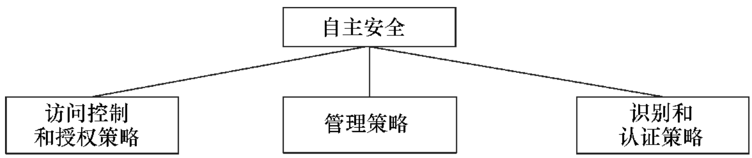


图 18.2　自主安全

查询修改算法：

输入：查询、安全约束

输出：修改查询

对于与查询相关的约束，请通过“否定”修改查询的where子句。

例如：假设不向Jane发放薪水，如果Jane从雇员那里请求信息，则修改查询将从雇员处获取的属性是没有工资。

重复该过程，直到处理所有相关的约束。

最终结果是修改查询。

图 18.3　查询修改

18.2.2　推理问题

推理是构成查询的过程，能从收到的合法响应中，推断出未经授权的信息。推理问题存在于所有类型的数据库系统中，并已在多级数据库的环境中得到了广泛的研究。推理问题的早期发展集中在统计数据库的安全性上。然后，重点就是用安全约束进程来解决推理问题。研究人员还使用概念结构，来设计数据库应用程序，并在设计时通过推理来检测是否有安全违规。推理有很多技术性挑战，其中包括不可抵赖性和问题的复杂性。推理问题的发展如图 18.4 所示。最近，推理问题因为隐私内容而受到很多关注。数据

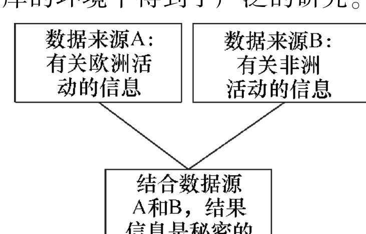


图 18.4　推理问题的相关方面

挖掘等技术被广泛用于国家安全，这引起了对隐私问题的关注。隐私问题是推理问题的一种形式，可以从公共信息中推导出高度私密的信息。

18.2.3　分布式/异构数据管理安全

分布式、异构和联合数据库系统的安全性，对于许多操作环境至关重要。例如，个人的数据管理系统可以执行自己的策略。这些策略必须结合起来形成全局策略。策略整合是异构和联合数据管理系统中的一个主要挑战。图18.5为策略整合。

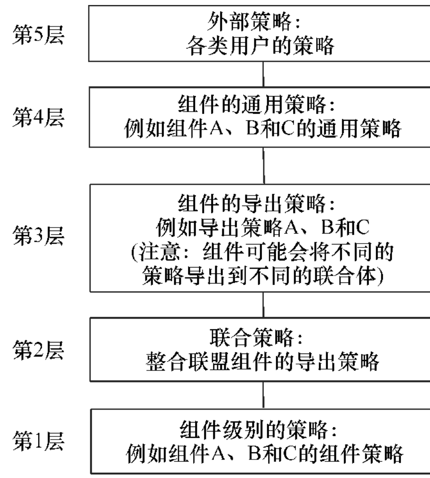


图18.5　策略整合

18.2.4　对象数据系统安全

对象技术对于许多应用程序很重要，包括编程语言、应用程序、系统的设计和分析、互连和数据库等。例如，Java等编程语言是基于对象的。分布式对象管理系统连接异构数据库和应用程序，数据库和应用程序使用对象来建模。可以使用对象组件创建大型系统。最后，对象技术在建模和设计方面是非常受欢迎的。因此，对象的安全性至关重要。也就是说，我们需要安全的对象编程语言、安全的对象数据库、安全的分布式对象系统和安全的对象组件，并使用对象来建模安全的应用程序。在本章参考文献［THUR05］中，我们讨论了与数据库和应用程序相关的各种类型的对象安全技术。例如，在安全的对象模型中，访问可以被控制到对象实例、属性、方法和类。图18.6为对对象的访问控制。

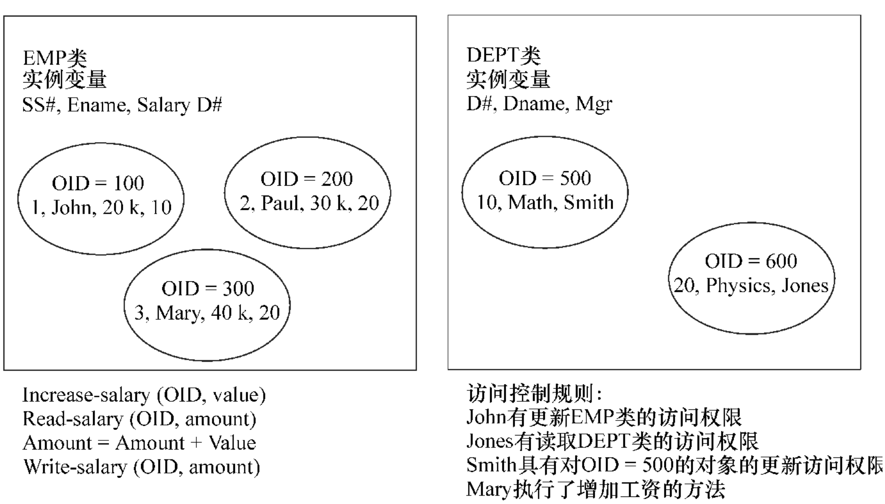


图18.6　对对象的访问控制

18.2.5 数据仓库、数据挖掘、安全和隐私

许多组织正在开发数据仓库。仓库对不同的用户，本质上是提供了不同的数据视图。例如，一家公司的总裁可能希望看到季度销售数据，而部门经理则可能希望看到每日销售数据。这些数据仓库必须是安全的。图 18.7 为数据仓库的安全方面。例如，基于个人数据库需要安全策略，我们也必须开发针对数据仓库的策略。

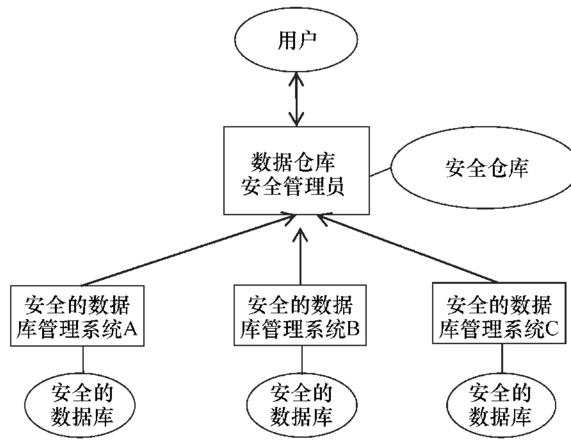


图 18.7 数据仓库安全

在本章参考文献［THUR05］中，我们讨论了数据挖掘与安全性之间的关系。例如，数据挖掘可以用于处理如入侵检测和审计之类的安全问题。另一方面，数据挖掘也加剧了推理和隐私问题。这是因为用户可以使用各种数据挖掘工具并结合不同的信息，来推出可能是敏感和私有的新信息，如图 18.8 所示。最近就有大量的侵犯隐私的讨论，就是由于数据挖掘而导致的。我们在本章参考文献［THUR05］中讨论了隐私问题以及基于隐私保护的数据挖掘的概念。

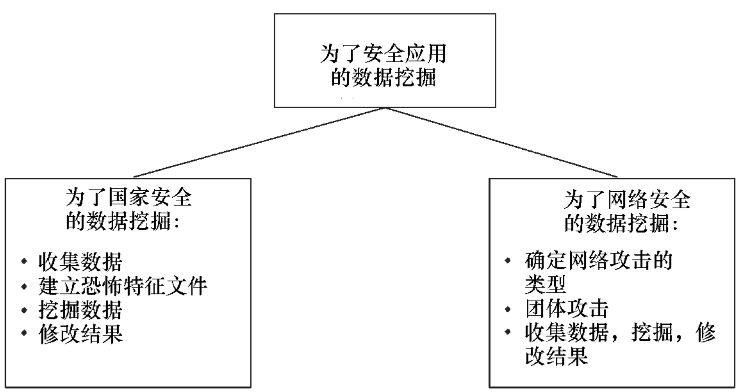


图 18.8 数据挖掘和安全性

18.2.6 信息管理安全

安全性对信息管理系统的功能影响很大，比如多媒体信息管理和工作流/协同信息管理等。例如，考虑多媒体数据管理器中的查询操作。查询处理器必须审查访问控制规则和安全性，并相应地修改查询。如果已存在的操作 X 被分类，则该查询不能被发送

到如摄像机等未分类的多媒体数据收集器中来拍摄该事件。安全性也会对多媒体的编辑和浏览产生影响。当浏览多媒体数据时，系统必须确保用户具有正确的访问权限，以浏览链接或访问与链接相关联的数据。在多媒体被编辑的情况下，当将不同级别的对象组合起来形成影像时，必须相应地对影像对象进行分类。我们可能需要把构成影像的各类帧，或者不同等级的高亮标记归类。此外，当影像被编辑（如删除电影的某些部分）时，则需要重新编辑对象的级别。

接下来，我们考虑工作流系统，其目的是让用户具有正确的凭证来执行特定任务。例如，在为一个项目进行采购的情况下，首先，只有项目负责人可以发起请求，秘书随之键入请求。然后，管理员必须使用他/她的信用卡进行采购，最后，收发室有交付的权力。在协作环境中，信任与协商发挥着重要作用。例如，各方如何相互信任来解决问题？例如 A 给 B 一些信息，假设 A 和 C 互不通信，B 可以与 C 共享信息吗？我们还需要安全的数据管理技术来管理工作流和协作应用程序的数据。图 18.9 说明了安全协作。

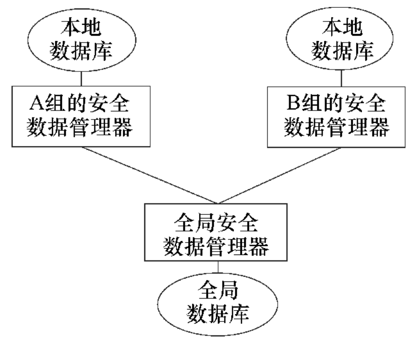


图 18.9　安全协作

18.2.7　知识管理安全

知识管理就是企业共享资源和专业知识，以及建立知识资本，从而增加竞争力。知识管理的挑战之一就是维护安全。商业机密必须高度保密，防止竞争对手访问。这就意味着需要执行某些形式的访问控制，例如基于角色的访问控制、凭证机制或加密。

为了拥有安全的知识管理，我们需要有安全的策略、流程和指标（见本章参考文献［BERT06］）。也就是说，指标必须包括对安全相关信息的支持。流程必须包括安全操作。策略必须包括安全策略。当创造知识时，创造者可以指定谁可以转让知识。附加的访问控制技术可能由知识管理者执行。知识共享和知识转移操作，同时还必须执行访问控制和安全策略。安全的知识管理架构可能围绕公司的内联网建立。图 18.10 说明必须保护的各种信息，以确保安全的知识管理。

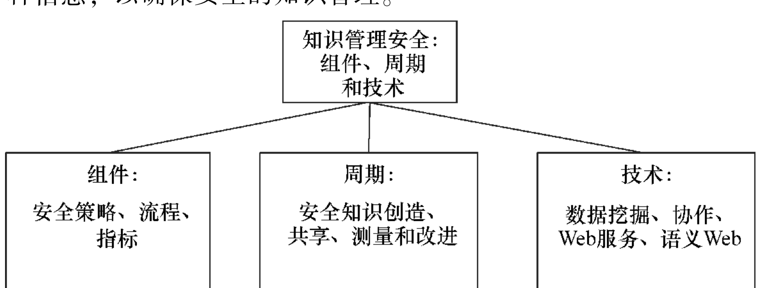


图 18.10　知识管理安全

18.3　云的影响

在本节中，我们将讨论云对18.2节讨论的数据管理功能安全的影响。

18.3.1　自主安全

云数据管理安全的大部分工作都集中在云查询处理安全上。例如，在我们开发的原型中，策略用XACML表示。该查询根据策略进行修改，并在云上执行。我们利用Hadoop/Map/Reduce框架，以及分布式存储管理和关系数据管理的Hive框架。我们还对XACML中指定的查询实施了查询重写，并使用云上的SPARQL查询处理器来实现查询语义Web数据。查询处理器托管在Hadoop/Map/Reduce框架上。查询处理的挑战是利用云提供的分布式处理和资源利用能力。我们还在云上实施了基于XACML的策略引擎，用于查询处理安全。这个策略引擎在本章参考文献［HUSA11］中讨论。图18.11说明了策略引擎流程。

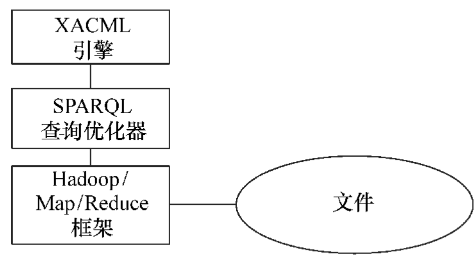


图18.11　XACML策略引擎

18.3.2　推理问题

关于推理问题，云平台提供了理想的解决方案。为了处理推理问题，必须处理包含历史信息和外部知识在内的大量数据。如此大量的数据可以在云中存储和处理。如前所述，我们已经开发了一个用于处理推理问题的基于RDF的策略引擎。这个策略引擎现已在云上运行（见本章参考文献［CADE12］）。图18.12为我们的RDF策略引擎。

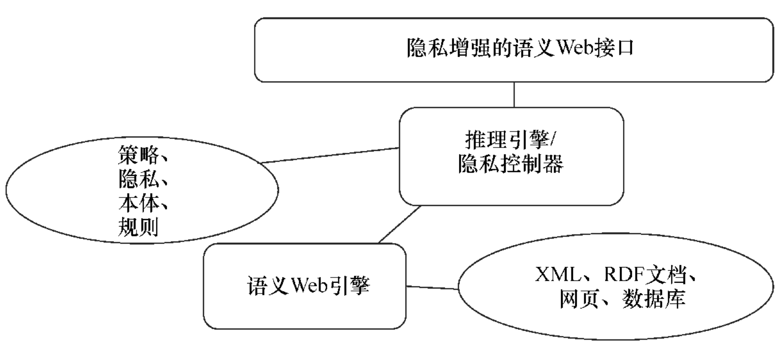


图18.12　RDF策略引擎

18.3.3　分布式和异构数据管理安全

由于云计算实质是基于分布式计算，因此自然就适用于分布式数据管理。例如，数

据和处理是分布在云端的。对于集成异构数据库，需要匹配模式以及对齐本体。例如，对于解决实体问题，我们已经将模式匹配和本体对齐作为解决方案并进行了探索。这些模式匹配和本体管理算法已经在云上实现（见本章参考文献［ALIP11］）。这里面临的挑战是在云上实施策略集成算法。在我们的原型之中，我们假设每个组织将其数据和策略存储在云端。各个组织将根据策略共享数据。

18.3.4 对象系统安全

执行在对象模型上的策略，可以与处理关系系统的查询重写采用同样的方法。对于集成各种组件的对象，如对象管理组（Object Management Group，OMG）。这样的组织正探索在云上实现对象请求代理。为这些系统开发的安全属性必须在云上实现。

18.3.5 数据仓库、数据挖掘、安全和隐私

Oracle 和 IBM 等数据管理公司正在云端实现数据仓库产品。这里面临的挑战是在云端数据仓库中实现要执行的策略。云端数据挖掘受到多方关注，数据挖掘面临的一个挑战是确保个人的隐私。过去十年中，隐私保护数据挖掘算法也已经出现。现有许多解决方案，如多方计算，都可以进行密集处理，也是云实现的理想选择。

18.3.6 信息管理安全

信息管理安全相关的应用程序，如内部威胁检测和信息共享正在云上实施。此外，在云上管理多媒体数据已经受到多方关注。多媒体服务已经在 PaaS 和 SaaS 之间实现。这样的云被称为多媒体云。这些云的移动应用所提供的服务质量，以及流式视频，将安全服务与多媒体服务相结合仍是一个挑战。

在 PeopleSoft 和 SAP 等企业资源计划系统中，已经实施了工作流应用程序。这些系统还实现了访问控制策略，例如基于角色的访问控制。随着 Oracle 和 SAP 等公司将其数据和应用程序转移到云端，我们可以期待在云上提供安全的工作流处理。在合作方面，个人或团体合作来解决问题，他们可以利用云将数据和策略存储在云端。

18.3.7 知识管理安全

安全的知识管理通常作为 Web 服务的集合来提供。例如，组织可能希望找到一个特定项目的专家。专家查找此类 Web 服务可以参与这个工作，该服务可以通过企业内部网找到专家。这些 Web 服务可以在云上实现，以提高性能。也就是说，知识管理服务可以实现为 SaaS。

18.4 总结和展望

在本章中，我们讨论了安全的数据管理功能，然后描述了云对这些功能的影响。特别是，我们讨论了自主安全性、多层次安全性、推理问题、对象安全、多媒体系统安

全、分布式数据管理安全、异构数据集成安全、数据仓库安全、数据挖掘、信息管理安全和知识管理安全。由于存储、挖掘以及挖掘大量信息的需要，因此保护用于管理数据、信息和知识的云将继续成为 IT 中的关键领域。

我们的工作重点主要是开发用于云数据管理系统的原型。我们还开发了云中安全信息共享系统。我们将在本书第六部分和第七部分中，讨论为云数据管理安全而开发的一些原型。

参考文献

[ALIP11] Alipanah, N., P. Parveen, L. Khan, and B. M. Thuraisingham, Ontology-driven query expansion using map/reduce framework to facilitate federated queries. *In Proceedings of International Conference on Web Services* (ICWS 2011), Washington, DC, 712–713.

[BERT06] Bertino, E. et al., Secure knowledge management, *IEEE Transactions on Systems, Man and Cybernetics*, May 2006.

[CADE12] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. M. Thuraisingham, A cloud-based RDF policy engine for assured information sharing. *In Proceedings of ACM Symposium on Access Control Models and Technologies* (SACMAT 2012), Newark, NJ, 113–116.

[HUSA11] Husain, M. F., J. P. McGlothlin, M. M. Masud, L. R. Khan, and B. M. Thuraisingham, Heuristics-based query processing for large RDF graphs using cloud computing. *IEEE Trans. Knowl. Data Eng.*, 23(9), 1312–1327, 2011.

[THUR05] Thuraisingham, B., *Database and Applications Security*, CRC Press, Boca Raton, FL, 2005.

第 19 章　云计算安全指南

19.1　概述

在本章中，我们将概述云计算安全性的指南。这些指南是由 NIST 制定的，实际上是总结了 NIST 文件中的论述。有关指南的更多细节，我们希望读者去参考本章参考文献［NIST］。为了实现 NIST 所讨论的指南，我们将云计算、云计算服务模型、部署模型和安全问题的定义纳入其中。

NIST 讨论的准则涵盖了几个主题。这些主题包括服务和部署模型、架构管理、数据保护、安全和隐私、可用性以及事件响应。由于云计算安全仍在发展，读者应该注意到这些指导方针也将随之发展。然而，我们已经发现这些指南对于了解安全的云计算来说是非常有用的。该指南也讨论了服务和部署模型、身份管理以及可用性。

本章的结构如下。将在 19.2 节中对指南进行概述。在 19.3 节对本章进行总结。图 19.1 为本章讨论的指南包含的内容。云计算安全的概述可以在本章参考文献［MATH09］中找到。有关云计算安全的白皮书已经发布在云安全联盟（见本章参考文献［CSA］）的网站上。

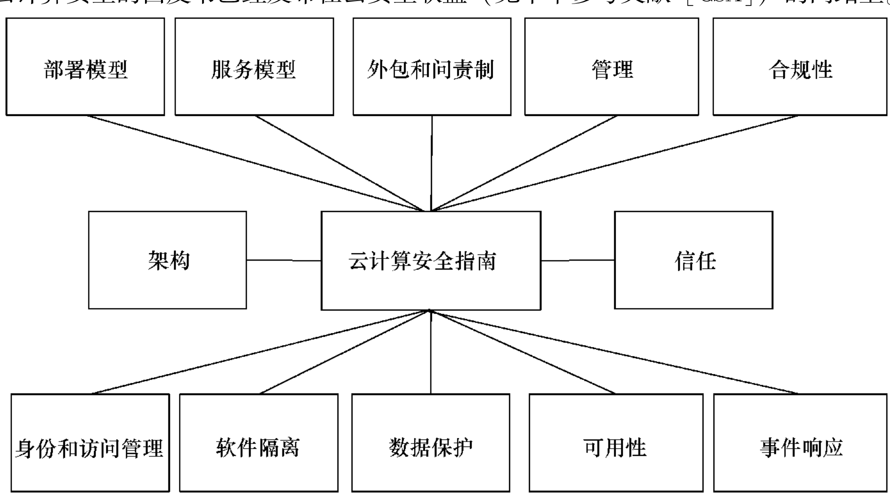


图 19.1　云计算安全指南

19.2　指南

定义：云计算已被 NIST 定义为一种无处不在、方便的、按需访问网络中共享的可

配置计算资源池（例如，网络、服务器、存储、应用程序和服务）的模型，可以通过最少的管理努力或与服务提供商的交互来快速提供和发布（见本章参考文献［NIST］）。

部署模型：如本章参考文献［NIST］所述，有多种部署模型。

公有云是通过互联网，为公众提供基础设施和计算资源的云。它由云提供商所拥有和运营，为消费者提供云服务，并且对于客户组织来说是外部的。私有云是计算环境仅针对单个组织运行的云。它可以由组织或第三方管理，并且可以在一个组织的数据中心内或其外部托管。私有云也可以使一个组织更好地控制云的资源和操作。混合云涉及两个或多个云的组合，其中至少一个是公有的，一个是私有的。社区云是由被称为社区的组织集合管理的云。

服务模型：如 NIST 所述，服务模型规定了组织对云资源的控制。这里有 3 种主要的服务模型。

软件即服务（Software – as – a – Service，SaaS）是一种服务交付模型，它由云服务提供商提供，指的是所有运行在云上的应用程序。这些应用可能包括金融应用和医疗应用。这个模型中的安全性由云服务提供商提供。平台即服务（Platform – as – a – Service，PaaS）是提供计算平台的服务交付模型，可以开发和部署应用程序。就是说，编程工具和数据库是 Platform 提供的关于服务的例子。云提供商和云消费者之间的安全规定是分开的。基础设施即服务（Infrastructure – as – a – Service，IaaS）是一种服务，能提供模型、服务器、软件和网络设备的基本计算基础设施，并作为可以托管其平台和应用程序的服务来提供。

外包和问责制：通过云计算，组织可以外包其应用程序、数据和处理。如 NIST 所述，由于外包，所以存在对安全和隐私问题的担心。例如，与数据迁移到公有云相关的风险是什么？

公有云有 3 种类型。第一种不需要消费者承担任何成本，并通过广告获得收入。第二种公有云收取费用，但不包括任何广告。第三种也是基于费用的，但是这些费用是在消费者和提供商之间协商决定的。

NIST 表示，一个组织最终要对其数据和处理的安全性负责，每种类型的公有云都会带来不同的安全挑战。

管理：我们在第 18 章讨论了管理。在这里，我们将讨论由 NIST 所定义的管理。

管理是关于组织对应用开发、设计、实施、测试、使用和监控服务的策略、程序和标准的控制和监督。

这里的挑战是为云服务开发适当的管理方法。这将涉及确定云服务提供商和组织的角色和职责，以及确定风险、引导风险分析流程、管理和减轻风险。

合规性：如 NIST 所述，合规是指组织有责任按照既定的法律、法规、标准和规范进行运作。由于云跨越组织边界，在许多情况下跨越多个州，不同的法律必须进行不同的分析和执行。因此，数据安全和隐私成为关键问题。另外，还需要适当的审计机制，以确保法规和法律得到适当执行。

信任：根据 NIST 所述，通过云计算，组织放弃对其数据和处理的控制，因此大多

数安全和隐私控制都由服务提供商掌握。因此，组织必须信任服务提供商。但是，要有保护敏感数据的规定。例如，敏感数据通常必须加密，因此不能放在公有云中。在云计算模式中也存在可能的内部威胁，因为服务提供商将可以访问资源和数据。在云中移动数据也增加了安全隐患。数据所有权必须在组织和服务提供商之间建立起来。例如，如果用户将数据存储在云中，那么谁将拥有这些数据？云服务还可以转换成多个服务。因此，必须检查服务组合的安全问题（见本章参考文献［THUR10］）。这里的另一个挑战是保护元数据。这将涉及有关用户的信息、账户联系信息和各种类型的合同。这些元数据可能会侵犯用户隐私。

风险：与任何 IT 系统一样，基于云服务的风险管理至关重要。如 NIST 所述，风险管理是识别和评估组织操作、组织资产或信息系统个体操作后果的过程，并采取必要的步骤将其降低到可接受的水平。

架构：在本节中，我们将讨论 NIST 指南给出的架构问题。虚拟机是云的核心。如 NIST 所述，虚拟机通常用作 IaaS 云的抽象单元，并与云存储架构进行松散耦合。

管理程序（也称为虚拟机监视器）是用于操作多租户虚拟机（称为客户操作系统）的操作系统和硬件平台之间的附加软件层。这些虚拟机管理程序容易遭受攻击。如 NIST 所述，大多数虚拟化平台都有能力创建基于软件的交换机，并进行网络配置，它们都是虚拟环境的一部分，目的是允许同一主机上的虚拟机更直接、更高效地进行通信。

这些虚拟网络也可能受到攻击。最后，要说的是，云中的客户端和服务器都必须被保护。

身份和访问管理：身份管理确保用户只能访问授权信息。这里的挑战是为云开发适当的访问控制程序。如 NIST 所述，身份联盟允许组织和云提供商信任，并共享数字身份以及跨组织和服务提供商域的属性，并提供单点登录的手段。

我们目前正在研究各种标准，如 SAML 和 OpenID。认证是建立用户身份的过程，SAML 正在成为云端的流行认证标准。SAML 请求和响应消息通常通过 SOAP 来映射，SOAP 依赖于 XML 的格式。SOAP 消息以数字方式签名。访问控制确保用户可以访问他/她被授权访问的资源。另外，用于访问控制的 XACML 标准在云提供商中变得越来越流行。事实上，我们的许多实验系统已经为各种类型的数据管理器实现了基于 XACML 的访问控制。

软件隔离：由于多个用户（即租户）共享云，因此进程隔离至关重要。如 NIST 所述，在 IaaS 的云计算环境中，多租户往往通过复用来自潜在的不同消费者的虚拟机执行体来完成相关任务，这些消费者是同一物理服务器上的。在客户虚拟机上部署的应用程序可能会受到攻击。PaaS 和 SaaS 中的多租户软件隔离也正在探索中。关键要素是管理程序的安全性。虚拟机管理程序是在客户操作系统之间提供隔离的。因此，如果管理程序遭到攻击，则整个云可能会受到影响。NIST 文件指出基于虚拟机的云基础架构的多租户将带来新的威胁。最严重的威胁是，恶意代码可以逃避其虚拟机的限制，并干扰虚拟机管理程序或其他客户虚拟机。

数据保护：存储在公有云中的数据位于共享环境中，数据可能属于多个客户。将敏感数据存储在公有云中是一个重大挑战。典型的攻击包括对加密协议的攻击和对口令的攻击。还有谁拥有数据的问题，是创建数据的组织还是云服务提供商？一般情况下，拥有数据的就是组织。因此，组织有责任确保数据安全和隐私。NIST 已经制定了数据安全技术，包括数据隔离、适当的访问控制和数据清理。在组织删除数据后，服务提供商必须确保云中没有残留的数据。

可用性：云的可用性对于客户很重要。服务提供商必须在 SLA 中指定可用性细节。有时可能会暂时中断云或长时间中断云。在这种情况下，需要适当地备份程序，以使服务不会中断，还有避免服务提供商停业的风险。最后，云可能会遭到拒绝服务攻击。因此，这些攻击必须在造成损坏之前被检测出来。

事件响应：如 NIST 所述，事件响应包括用于处理针对计算机系统的安全性受到攻击造成后果的有效系列方法。云服务提供商有责任确保系统运行，包括应用程序、网络操作系统和数据库系统。必须检查云的日志文件，以便可以恢复系统。日志数据必须被及时提供。服务提供商必须与组织以及事件分析和响应小组合作，以确定问题、取证分析和解决问题。

19.3 总结和展望

本章对 NIST 所讨论的云计算安全指南进行了概述。指南包括部署模型、Web 服务及管理、合规性、可用性、身份和访问管理、架构、事件响应、信任、软件隔离和数据保护。我们基本上总结了 NIST 提供的各种指南。更多详细信息，请参见本章参考文献［NIST］。应该指出，云计算将会不断发展，随之安全问题也会发展。因此，这些指南将继续演进。然而，这些指南是部署安全的云的一个很好的起点。

现在我们已经提供了云安全问题的概述以及云计算功能安全，重点是云数据管理安全，我们将在第 20 章中对使用云执行安全功能以及云计算产品和服务中的各种安全措施进行概述。

参考文献

[CSA] Cloud Security Alliance, https://cloudsecurityalliance.org/

[MATH09] Mather, T., Kumaraswamy, S., and Latif, S., *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, O'Reilly, Sebastopol, CA, 2009.

[NIST] Guidelines on security and privacy in public cloud computing, *National Institute of Standards and Technology*, http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909494

[THUR10] Thuraisingham, B., *Secure Semantic Service-Oriented Systems*, CRC Press, Boca Raton, FL, 2010.

第 20 章　安全即服务

20.1　概述

第 16 ~ 19 章讨论了云计算的安全问题。例如，讨论了有关保护云基础设施的各个方面。一方面云计算技术必须是安全的，另一方面云计算也可用于安全服务。这包括为恶意软件检测服务、垃圾邮件检测服务和数字取证服务而进行数据挖掘。将云应用于安全目的已经被称为“安全即服务”。在本章中，我们将讨论云提供安全性作为服务的示例。我们的主要重点是数据挖掘服务。第 21 章将对云计算安全产品（包括安全作为服务的产品）进行概述。

数据挖掘在安全性方面有许多应用，包括国家安全（例如监控）以及网络安全（例如病毒检测）。对国家安全的威胁包括攻击建筑物、破坏关键基础设施，如电网和电信系统（见本章参考文献［BOLZ05］）。我们可以通过研究数据挖掘技术，以找出可疑人员和有能力进行恐怖活动的人。数据挖掘也被应用于提供入侵检测和审计等解决方案。在本章中，我们将主要关注用于网络安全应用的数据挖掘。

要理解应用于维护国家、计算机和网络的机制，我们需要知道威胁的类型。在本章参考文献［THUR03］中，我们描述了实时威胁以及非实时威胁。对于实时威胁是，必须在一定时间内采取行动，以防止一些灾难性局势的威胁。需要注意的是，非实时威胁可能会随时间的变化而变成实时威胁。例如，可以怀疑一群恐怖分子最终会做出一些恐怖主义的行为。但是，当我们设置时间范围时，如在 2004 年 7 月 1 日之前，威胁可能会发生，之后它就会成为一个实时威胁，我们必须立即采取行动。如果时间范围更窄，如“两天内威胁会发生”，那么我们不能容忍威胁响应中出现任何的错误。

我们在将数据挖掘应用到国家安全和网络安全方面已经做了很多工作。然而，数据挖掘技术是计算密集型的。因此，安全的数据挖掘服务是在云中实施的最好选择。

本章的结构如下。在 20.2 节中，我们将讨论网络安全应用的数据挖掘服务。特别是，我们将讨论对计算机和网络的威胁，并描述用数据挖掘的应用程序来检测这些威胁和攻击。我们目前在 UTD 的一些研究在 20.3 节中进行讨论。20.4 节将讨论属于安全服务类别的其他一些服务。本章总结在 20.5 节。图 20.1 说明了本章讨论的概念。更多细节也可以在本章参考文献［THUR04］中找到。



图 20.1　安全即服务

20.2 用于网络安全的数据挖掘服务应用

20.2.1 简介

在本节中，我们将介绍为了安全的数据挖掘服务。首先，我们将概述各种类型的威胁，包括网络恐怖主义以及通过访问控制、特洛伊木马和病毒的安全违规行为。然后，我们将描述数据挖掘技术作为检测安全违规的服务，如何开发。以下是数据挖掘服务如何工作的一个例子。当组织想要检查他们的网络是否被入侵时，他们将调用云数据挖掘服务来处理其审计数据。数据挖掘服务将确定是否存在任何入侵。在本书的第七部分中，我们会论述入侵检测和恶意软件检测的数据挖掘服务。

本节的结构如下。在 20.2.2 节中，我们概述网络恐怖主义，然后将讨论内部威胁和外部攻击。恶意入侵是 20.2.3 节的主题。20.2.4 节讨论信用卡和身份信息窃取。对关键基础设施的攻击将在 20.2.5 节中讨论。20.2.6 节将讨论用于网络安全的数据挖掘。图 20.2 为网络安全威胁。

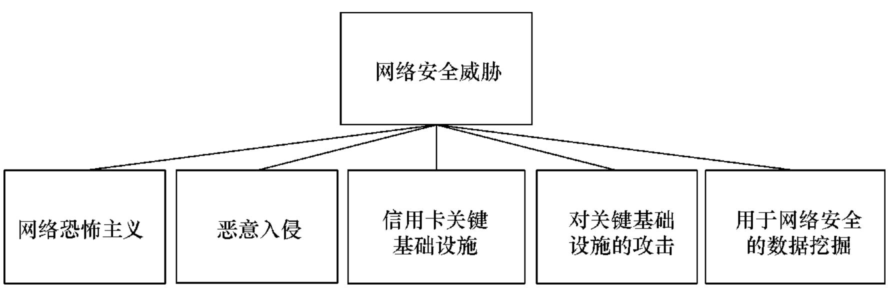


图 20.2 网络安全威胁

20.2.2 网络恐怖主义、内部威胁和外部攻击

网络恐怖主义是今天对我们国家构成的重大恐怖威胁之一。正如我们前面提到的那样，现在有大量的信息可以通过网络获得。对我们的计算机以及网络、数据库和互联网的攻击，可能会给企业造成损失。据估计，网络恐怖主义可能会给企业造成数十亿美元的损失。例如，考虑一个银行信息系统。如果恐怖分子攻击这样一个系统，耗尽资金账户，那么银行可能会损失数百万美元，甚至数十亿美元。通过使计算机系统瘫痪，几小时数百万的生产力可能会丢失，最后就相当于同等金钱的流失。即使在一些事故中，简单的停电也可能导致几小时的生产力下降，从而导致重大的经济损失。因此，我们的信息系统必须是安全的。我们讨论各种类型的网络恐怖袭击。一个是传播病毒和特洛伊木马，其可以删除文件和其他重要的文档；另一个则是入侵计算机网络。

需要我们注意的是，威胁可能发生在外部或组织内部。外部攻击是组织以外的人对计算机的攻击。我们听到黑客入侵计算机系统，并在组织内造成严重破坏，就是外部攻击。有黑客开始传播病毒，这些病毒对计算机系统造成极大的伤害。但更为险恶的问题

就是内部威胁。组织内部有人研究了业务实际，制定了削弱组织信息资产的计划。这些人可能是正规员工，甚至是在计算机中心工作的人员。这个问题是相当严重的，因为有人可能伪装成内部人员去造成各种损害。在本书接下来的几个部分中，我们将研究数据挖掘如何检测并尽可能防止这种攻击。

20.2.3　恶意入侵

恶意入侵可能包括侵入网络、Web 客户端和服务器、数据库和操作系统。许多网络恐怖主义袭击是由于恶意入侵造成的。我们听到很多关于网络入侵的情况。这里发生的是入侵者试图利用网络，并获得正在传输的信息。这些入侵者可能是人类入侵者或人们制造的特洛伊木马。文件也可能发生被入侵的情况。例如，可以伪装成别人，登录别人的计算机系统并访问文件。入侵也可能发生在数据库上，扮成合法用户的入侵者可能会执行 SQL 查询和访问他们未被授权查看的数据。

基本上，网络恐怖主义包括恶意入侵，以及通过恶意入侵或其他方式的破坏。网络安全包括试图为防止网络攻击或网络恐怖主义提供解决方案的安全机制。当我们讨论恶意入侵或网络攻击时，可能需要考虑非网络世界，即与非信息相关的恐怖主义，然后将这些攻击转化为对计算机和网络的攻击。例如，一个小偷可以通过一个后门进入建筑物。以同样的方式，计算机入侵者可以通过某种后门进入计算机或网络，这个后门是由恶意内部人员在无人值守的情况下故意构建的；另一个例子是小偷用面具进入银行偷钱。这里的类比是入侵者伪装成别人，合法进入系统，并取走所有的信息资产。这里的现实世界中的钱将转化为网络世界的信息资产。也就是说，非信息相关攻击和信息相关攻击之间有许多相似之处。然后，我们可以着手制定针对这两种攻击的对策。

20.2.4　信用卡欺诈和身份盗用

这些天我们听到有关信用卡欺诈和身份窃取的情况。在信用卡欺诈的情况下，其他人持有一张别人的信用卡，进行各种购买行为，当卡的所有者发现的时候，可能为时已晚。那时，小偷可能已经离开了这个国家。

身份盗用是更严重的犯罪行为。在这里，将自己假设为另一个人，通过掌握社会保障号码，并以其他人的名义，执行所有交易。这种交易甚至可能是出售房屋，并将收入存进用于欺诈的银行账户。当业主发现时，为时已晚。由于身份被盗用，业主很有可能损失数百万美元。

我们需要探索使用数据挖掘来进行信用卡欺诈检测和身份盗用检测。在检测信用卡欺诈方面已经做了一些努力（见本章参考文献［CHAN99］）。我们需要积极开展检测和防止身份盗用工作。

20.2.5　对关键基础设施的攻击

对关键基础设施的攻击可能会削弱一个国家及其经济实力。基础设施攻击包括攻击对国家运作至关重要的电信线路、电子器件、电力、燃气、水库、供水、食品供应以及其他基础设施。

对关键基础设施的攻击可能发生在任何类型的攻击之中，无论它们是非信息相关的或信息相关的，还是生物恐怖主义攻击。例如，可以攻击运营电信业的软件，并关闭所有电信线路。同样，运行电力和天然气供应的软件也可能受到攻击，也可以通过炸弹和爆炸物进行攻击。也就是说，电信线路可能会被炸弹袭击。攻击高速公路和铁路等运输线路也是对基础设施的攻击。

基础设施也可能受到飓风和地震等自然灾害的袭击。我们的主要关注点是那些通过恶意攻击对信息和非信息相关的基础设施进行的攻击。我们的目标是研究数据挖掘和相关数据管理技术，以检测和防止这种对基础设施的攻击。图 20.3 说明了对关键基础设施的攻击。

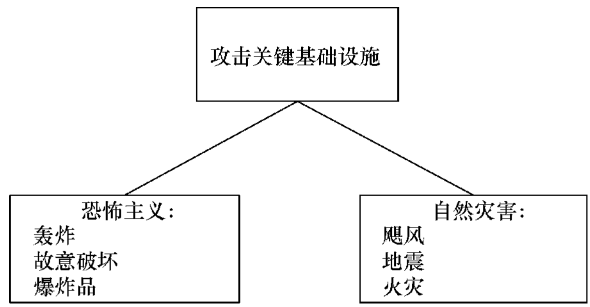


图 20.3　对关键基础设施的攻击

20.2.6　用于网络安全的数据挖掘服务

数据挖掘正在应用于解决入侵检测和审计等问题。例如，异常检测技术可用于检测异常模式和行为。链接分析可用于追踪病毒制造者。分类可用于对各种网络攻击进行分组，然后使用分析器在发生攻击时检测攻击。预测可能将被用来决定未来的攻击，这取决于通过电子邮件和电话交谈来了解恐怖分子的信息。此外，对于某些威胁，非实时数据挖掘可能就足够了，而对于某些其他威胁（例如网络入侵），我们可能就需要实时数据挖掘。许多研究人员正在研究使用数据挖掘进行针对入侵的检测。虽然我们需要某种形式的实时数据挖掘，也就是说，结果必须实时生成，我们还需要实时构建模型。例如，信用卡欺诈检测是一种实时处理的形式。然而，在这里，模型通常是提前建立的。实时建模仍然是一个挑战。数据挖掘也可用于分析 Web 日志以及分析审计跟踪。根据数据挖掘工具的结果，可以确定是否发生了任何未经授权的入侵和/或是否提出任何未经授权的查询。

用于网络安全的数据挖掘的其他应用包括分析审计数据。人们可以构建一个包含审计数据的资料库或仓库，然后使用各种数据挖掘工具进行分析，看是否存在潜在的异常现象。例如，某些用户组可能会在上午 3 点到 5 点之间访问数据库。也可能是这个小组正在上夜班，在这种情况下可能会有一个合理的解释。但是，如果这个小组也在上午 9 点到下午 5 点之间工作，那么这可能就是一个不寻常的事情；另一个例子是，一个人一直在下午 1 点到 2 点之间访问数据库，但是在过去的两天里，他一直在上午 1 点到 2 点访问数据库。这可能是一个不寻常的事件，就需要进一步的调查。

内部威胁分析也是来自于国家安全和网络安全方面的问题。也就是说，在公司工作的且被认为是可信赖的人，有可能进行间谍活动。同样，那些能够正当访问计算机系统的人，也可能会制造特洛伊木马和病毒。抓捕这样的恐怖分子比在组织之外抓捕恐怖分子要困难得多。一个人可能需要监视公司所有人，包括系统管理员的访问模式，看他们是否在执行网络恐怖主义活动。现在多个团体在进行一些研究将数据挖掘应用于这些应用。图 20.4 为用于网络安全的数据挖掘服务。

虽然数据挖掘可用于检测和防止网络攻击，但数据挖掘也加剧了一些安全问题，如推理和隐私问题。比如，利用数据挖掘技术，可以从合法的响应中推断出敏感关联的信息。有关隐私和数据挖掘的更多细节，请参见本章参考文献［THUR05］。

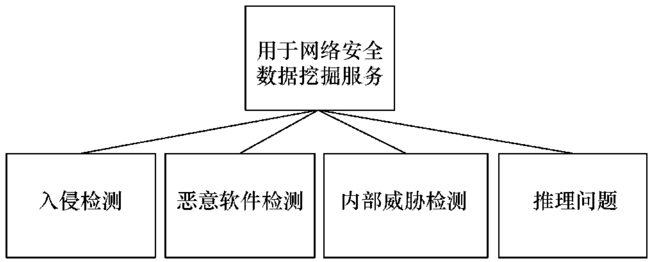


图 20.4　用于网络安全的数据挖掘服务

20.3　目前关于安全即服务的研究

我们正在 UTD 开发一些数据挖掘工具，用于网络安全目的。在第 19 章中，我们讨论了一个这样的入侵检测工具（见本章参考文献［AWAD09］）。可以将入侵定义为试图破坏资源的完整性、机密性或可用性的任何一组操作。随着系统变得越来越复杂，由于设计和编程的错误，或者通过使用各种“社交工程”的渗透技术，总是存在可利用的漏洞。计算机攻击分为两类：基于主机的攻击和基于网络的攻击。基于主机的攻击针对机器，并尝试访问该机器上的特权服务或资源。基于主机的攻击检测通常使用从审计过程中获取系统来调用数据的例程，该过程跟踪每个用户所做的所有系统调用。

基于网络的攻击通过故意占用或破坏网络资源和服务，来使合法用户难以访问各种网络服务。这种攻击可以通过发送大量的网络流量，利用网络服务中的已知故障，如通过使网络主机过载等来实现。基于网络的攻击检测使用网络传输数据（即 tcpdump）查看正在监视的机器上的通信流。入侵检测系统分为两类：异常检测系统和误用检测系统。

异常检测是基于偏离已建立的正常网络流量模式，来识别恶意流量的尝试。误用检测是基于恶意活动的已知模式识别入侵的能力。这些已知的模式被称为签名。异常检测能够捕获新的攻击。然而，新的合法行为也可能被误判为攻击，导致误报。目前最先进技术的重点是减少漏报和误报率。

我们最近在书中讨论的当前工具包括电子邮件蠕虫检测、恶意代码检测、流量检测和僵尸网络检测，以及分析防火墙策略规则（见本章参考文献［MASU11］）。图 20.5 说明了我们开发的各种工具。例如，对于电子邮件蠕虫检测，我们首先检查电子邮件并提取如“附件数量”之类的特征，然后使用如 SVM 或朴素贝叶斯分类器等技术来训练数据挖掘工具，并开发模型。最后我们测试模型并确定电子邮件是否有病毒/蠕虫。检测时，我们使用在各种网站上发布的训练培训和测试数据集。类似地，对于恶意代码检测，我们使用汇编代码和二进制代码来提取 n－gram 特征。我们首先使用 SVM 技术训练数据挖掘工具，然后测试该模型。分类器将确定代码是否是恶意的。为了对缓冲区进

行溢出检测，我们假设恶意消息包含代码，而正常消息包含数据。我们训练 SVM，然后测试，看消息是否包含代码或数据。有关在云中提供安全即服务的各种数据挖掘技术的讨论在本书附录 B 中给出。有些细节也可以在本章参考文献［AWAD09］中找到。

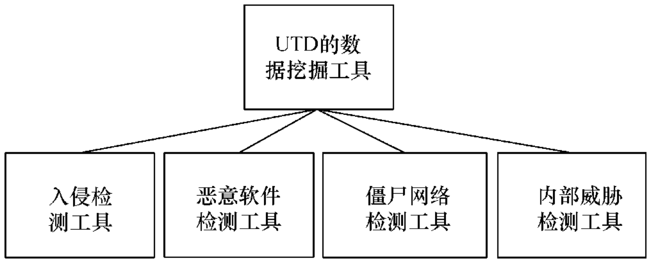


图 20.5　UTD 的数据挖掘工具

20.4　网络安全应用程序的其他服务

作为云端托管的其他候选服务包括电子邮件过滤、身份管理和数字取证。这些服务都是计算密集型的。在电子邮件过滤服务的情况下，组织可以将该服务外包给云端。云将检查电子邮件，可能会应用数据挖掘技术，并过滤出可疑的电子邮件。我们开发了用于电子邮件蠕虫检测的数据挖掘工具。在身份管理的情况下，组织将把识别和认证用户的处理外包给云端。需要注意的是，这不是云验证，而是用户必须通过身份验证才能访问云。这实际上是与如何访问组织的资源相关。然而，将用户认证留给云而不是组织。在进行取证方面，这些工具将在云端托管。组织将发送所有相关数据进行云取证。再次强调，这不是关于云取证，在这里只是通过云进行入侵检测。这是关于组织外包给云的取证活动。

实施安全服务的主要障碍就是安全本身。也就是说，这些由云实现的安全服务必须是安全的。例如，如果云要执行身份管理，则必须确保该进程是正确的。因此，将高度安全的功能外包给云仍然是一个挑战。图 20.6 说明了可以在云上实现的其他安全服务。

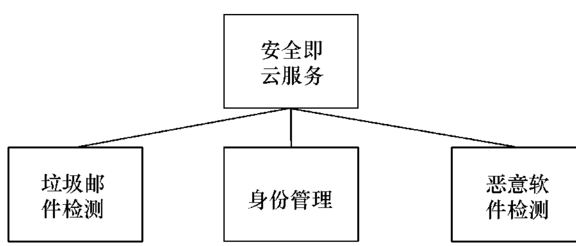


图 20.6　安全即云服务

20.5　总结和展望

本章我们讨论了用于安全的数据挖掘服务。因为这些数据挖掘服务是计算密集型的，因此它们在云计算上实施是理想的选择。主要执行安全功能的这些服务称为安全即服务。我们首先讨论了用于网络安全的数据挖掘，然后简要概述了我们正在开发的工

具。我们还讨论了一些其他服务，如电子邮件过滤、取证和身份管理。

接下来将讨论我们为安全性服务而开发的一些基于云的工具，这些包括云上的恶意软件检测、基于云的安全信息共享以及基于云的内部威胁检测。我们将在本书的第七部分讨论这些工具。第 21 章将讨论一些安全即服务的云产品。

参 考 文 献

[AWAD09] Awad, M., L. Khan, B. Thuraisingham, and L. Wang, *Design and Implementation of Data Mining Tools*, CRC Press, Boca Raton, FL, 2009.

[BOLZ05] Bolz, F., K. Dudonis, and D. Schulz, *The Counterterrorism Handbook: Tactics, Procedures, and Techniques, Third Edition Practical Aspects of Criminal & Forensic Investigations*, CRC Press, Boca Raton, FL, 2005.

[CHAN99] Chan, P., W. Fan, A. Prodromidis, and S. Stolfo, Distributed data mining in credit card fraud detection, *IEEE Intelligent Systems*, 14(6), 47–74, 1999.

[MASU11] Masud, M., L. Khan, and B. Thuraisingham, *Data Mining Tools for Malware Detection*, CRC Press, Boca Raton, FL, 2011.

[THUR03] Thuraisingham, B., *Web Data Mining Technologies and Their Applications in Business Intelligence and Counter-Terrorism*, CRC Press, Boca Raton, FL, 2003.

[THUR04] Thuraisingham, B., *Managing Threats to Web Databases and Cyber Systems, Issues, Solutions and Challenges* (eds: V. Kumar, J. Srivastava, Al. Lazarevic), Kluwer, MA, 2004.

[THUR05] Thuraisingham, B., *Database and Applications Security*, CRC Press, Boca Raton, FL, 2005.

第 21 章　云计算安全产品

21.1　概述

自从我们在 2000 年代中期开始知道云计算以来，就已经出现了无数的云安全解决方案。这些解决方案可以分为两类：第一类产品基本上为云提供安全保障。也就是说，这些产品可以保护云计算基础设施、平台或应用程序；第二类产品在云中提供安全服务。这些服务包括电子邮件安全服务、网页过滤服务和恶意软件检测服务。

虽然 Symplified 等一些公司已经开发出主要用于云计算的解决方案，但一些其他公司已经采用了云计算的安全解决方案。这些公司包括 IBM 等大型计算机公司。而一些其他公司，如 McAfee 和 Symantec 是通用安全解决方案的提供商，这些公司现在也提供在云中实施的安全解决方案。也就是说，它们提供安全即服务解决方案。

在本章中，我们提供各种云安全产品的概述。我们从本章参考文献［TECH］里进行的一项调查中获得了这份清单。我们对这些产品开展了调查。云安全产品的分类方法如图 21.1 所示。本章的结构如下。在 21.2 节中，我们将概述几个云安全产品，包括 IBM 和 Novell 等大公司，McAfee 和 Symantec 等安全解决方案公司，以及 Symplified 和 CloudPassage 等云安全公司。在 21.3 节对本章进行总结。

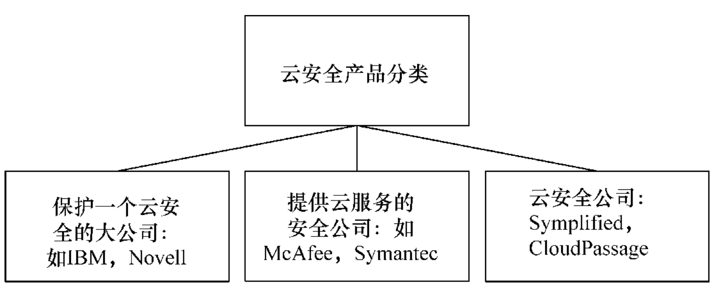


图 21.1　云安全产品分类

21.2　产品概述

Trend Micro（趋势科技）：网络安全软件及服务领域的全球领导者，总部位于日本东京和美国硅谷，提供包括防病毒和反垃圾邮件产品（见本章参考文献［TREND］）的安全解决方案。这些产品为网络、Web 和移动应用提供安全解决方案。该公司在 2008

年进入云安全领域，通过其产品“智能保护网络”（Smart Protection Network，SPN）。该产品的目标是保护客户端免受基于 Web 的恶意软件的侵扰。近年来，该产品已经扩展到安全云端，并提供广泛的云安全功能，包括虚拟化安全、云数据安全、策略管理和加密。

McAfee：McAfee 是全球最大的专业安全技术公司，为家庭用户以及企业和政府提供安全解决方案。其产品包括病毒扫描、电子邮件和网络安全以及加密。例如，McAfee 总体解决方案由各种安全工具组成，可防范恶意软件。最近，McAfee 为安全即服务提供了解决方案（见本章参考文献［MCAF］）。通过这项服务，客户可以获得基于云的数据保护、电子邮件保护和病毒扫描的解决方案。McAfee 现在是英特尔公司的一部分。

CA 科技：CA 科技公司的前身为国际计算机协会，是一家大型软件公司。它开发了系统软件，并在包括大型机和个人计算机在内的许多平台上开发了应用软件。通过收购多家公司，近几年来，它进入了云计算市场。他们的主要产品称为 CA Cloud Minder，为认证、身份管理和单点登录提供解决方案。

Symplified：虽然趋势科技和 McAfee 是安全解决方案公司，并且 CA 科技是一家软件公司，但 2006 年成立的 Symplified，目的是提供云安全解决方案。其主要内容是为云提供身份管理解决方案，包括使用 SAML 技术进行身份验证和单点登录（见本章参考文献［SYMP］）。

Symantec（赛门铁克）：赛门铁克以人工智能和数据库软件公司起步，然后发展到提供安全解决方案。它是最大的安全解决方案公司之一。它销售诺顿公司防病毒产品，并通过收购维尔软件提供存储解决方案。赛门铁克提供基于云的安全解决方案，包括病毒扫描、电子邮件和 Web 安全（见本章参考文献［SYMA］）。

Zscaler：Zscaler 成立于 2008 年，面向网络流量提供基于云的安全即服务。该产品为 Web 流量提供恶意软件保护，本质上是提供基于策略的 Web 访问。所提供的服务包括防病毒扫描、反垃圾邮件过滤、反间谍软件检测和 URL 过滤（见本章参考文献［ZSCA］）。需要注意的是，Symplified 为云提供身份和访问管理，Zscaler（如 Symatec 和 McAfee）在云中提供安全即服务。然而，与 Symatec 和 McAfee 不同，Zscaler 不是一般的安全解决方案公司。

Panda Security（熊猫安全）：熊猫安全是一家总部位于西班牙的安全公司，成立于 1990 年。其产品包括防病毒软件、垃圾邮件检测软件、防火墙应用和网络犯罪预防解决方案。最近，它提供基于云的安全解决方案。也就是说，它为电子邮件和互联网流量保护提供了安全服务（见本章参考文献［PAND］）。

White Hat（白帽）：白帽安全公司是作为企业和政府的安全解决方案的提供商。它还提供网站风险管理解决方案。2010 年，白帽加入了云安全联盟。最近，白帽提供了一个名为 Sentinel 的基于云的解决方案，为网站（见本章参考文献［WHIT］）提供了抗压安全。

CipherCloud：CipherCloud 为云提供安全解决方案。特别是，它为云提供加密解决方案，以便云客户可以将其数据安全地存储在云端。它为云系统提供安全解决方案，如

SalesForce.com、亚马逊 EC2、亚马逊 S3 和 Forcel.com（见本章参考文献［CIPH］）。需要注意的是，尽管白帽安全公司和 Zscaler 等公司为企业提供安全服务，但 Symplified 和 CipherCloud 可为云提供安全解决方案。

SecureAuth：SecureAuth 成立于 2005 年，为跨国企业开发身份提供商解决方案。根据本章参考文献［SECU］所述，该公司将单点登录与双因子身份验证相结合。最近，SecureAuth 为组织提供了作为云服务的身份管理解决方案。

CloudPassage：CloudPassage（见本章参考文献［CLOU］）成立于 2010 年，为云服务器提供安全性，包括虚拟化安全。其安全解决方案可以并入公有、私有和混合云。像 Symplified 一样，CloudPassage 的成立，为云提供安全保障。

Dome9 Security：如本章参考文献［DOME］所述，Dome9 Security 为云栈提供安全保护，以防黑客入侵。它提供了各种安全解决方案，包括策略管理、基于角色的访问控制、安全审计和强大的双因子身份验证。像 CloudPassage 一样，Dome9 Security 为云提供安全解决方案。

IBM：IBM 是最大的计算机公司，拥有大型机、数据库应用程序和安全解决方案等多种产品。其云安全解决方案包括数据保护、服务器安全以及策略管理（见本章参考文献［IBM］）。IBM 还提供用于保护云的咨询服务。

Novell：Novell 是一家非常大的软件和网络公司，为协作和网络提供了各种解决方案。Novell 的云安全解决方案称为 Novell 云安全服务（见本章参考文献［NOVE］），它可以通过 IaaS、SaaS 和 PaaS 级别的云提供商为组织提供身份管理解决方案。

TreatMetrix：如本章参考文献［THRE］所述，TreatMetrix 提供各种安全解决方案，包括①确定在线访问者是否合法；②恶意软件检测；③在欺诈和误用中保护移动应用程序。其基于云的解决方案通过云提供欺诈检测。就像熊猫安全，它通过云提供安全服务。

Okta：如本章参考文献［OKTA］所述，Okta 提供目录服务、单点登录、强身份验证、配置、工作流和内置报告。这些服务通过云提供，并与组织的身份管理服务相集成。SAP 和 Informatica 等公司使用 Okta 安全服务。

戴尔数据保护：戴尔提供企业范围的数据保护。它本质上是提供加密解决方案。如本章参考文献［DELL］所述，他们的产品保护系统磁盘和外部媒体，并且包含有强大的远程管理、审计和策略设置功能。

Awareness Technologies：Awareness Technologies 为员工监视和内部威胁保护提供安全解决方案。此外，其 InterGuard 产品套件还提供了 Web 扫描、数据丢失预防、笔记本电脑恢复监控、电子邮件和短信（见本章参考文献［AWAR］）的解决方案。它也是通过云提供安全服务。

HyTrust：HyTrust 为虚拟化软件提供安全保障。如本章参考文献［HYTR］所述，它在基于 VMware 的虚拟化基础设施的控制层面上实施策略。HyTrust 是虚拟设备，并为虚拟对象分配标签，随后在标签上执行策略。

Vyatta：如本章参考文献［VYAT］所述，Vyatta 为 Internet 协议提供 VPN 产品，以及虚拟防火墙和虚拟路由器。它的按需应变的网络安全解决方案正在纳入虚拟化、多核

的云计算平台。

StillSecure：如本章参考文献［STIL］所述，StillSecure 提供网络访问控制和安全托管服务。这些受管理的安全服务用来保护网络和数据。最近，StillSecure 推出了云监控服务产品，客户可以在其云平台下载和安装。

SafeNet：如本章参考文献［SAFE］所述，SafeNet 公司为各种企业提供数据保护解决方案。它本质上提供了加密技术。SafeNet 公司的云安全产品 ProcutV 为数据中心以及云中的数据提供加密解决方案。

Proofpoint：如本章参考文献［PROO］所述，Proofpoint 提供了安全即服务解决方案，用于保护企业数据。它还确保满足数据的管理、合规性和法规要求。提供的安全解决方案使组织能够将恶意内容从数据中去除，并防止窃取或丢失敏感数据。

Commtouch：Commtouch 是一家在德国成立的公司，提供电子邮件安全。它还提供防病毒保护，以及垃圾邮件和网络扫描的服务。如本章参考文献［COMM］所述，Commtouch 的全球视野云技术通过分析网络流量和电子邮件流量，来提供安全即服务。分析的结果被纳入到电子邮件和网络安全，以及防病毒保护的解决方案中。

21.3 总结和展望

本章概述了为云提供安全解决方案或为各种业务提供安全服务的各种商业产品。其中一些产品主要用于云端或在云端运行。一些其他解决方案已经采用了一般安全产品，并将其应用于云端。云的安全解决方案包括加密、虚拟化安全和云监控。安全服务包括身份管理、恶意软件检测、网络和电子邮件过滤。

我们选择了本章参考文献［TECH］提供的商业云产品调查中所列出的安全产品。不过，我们对这些产品展开了自身的调查。随着云技术的发展，我们预计产品数量将迅速增长。这些产品将提供云安全解决方案，或利用云提供安全服务。

参考文献

[AWAR] Awareness Technologies, http://www.awarenesstechnologies.com/
[CA] CA Technologies, http://www.ca.com/us/cloud.aspx
[CIPH] Cipher Cloud, http://www.ciphercloud.com/
[CLOU] Cloud Passage, http://www.cloudpassage.com/about/
[COMM] Commtouch, http://www.commtouch.com/#
[DELL] Dell Data Protection Solutions, http://www.dell.com/learn/us/en/04/campaigns/dell-data-protection-solutions
[DOME] Dome9 Security, http://www.dome9.com/about-us
[HYTR] Hytrust, http://www.hytrust.com/
[IBM] IBM, http://www-03.ibm.com/security/cloud-security.html
[MCAF] McAfee, http://www.mcafee.com/us/solutions/cloud-security/cloud-security.aspx
[NOVE] Novell Cloud Security, http://en.wikipedia.org/wiki/Novell_Cloud_Security_Service
[OKTA] Okta Service, http://www.okta.com/company/

[PAND] Panda Security, http://www.pandasecurity.com/usa/
[PROO] Proofpoint, http://www.proofpoint.com/about-us/index.php\
[SAFE] Safenet, http://www.safenet-inc.com/
[SECU] Secure Auth, http://www.gosecureauth.com/company/
[STIL] Still Secure, http://www.stillsecure.com/
[SYMA] Symantec, http://www.symantec.com/products-solutions/families/?fid=symantec-cloud
[SYMP] Symplified, http://www.symplified.com/
[TECH] Technavio Insights, Global Cloud Security Market, 2010–2014.
[THRE] ThreatMatrix, http://threatmetrix.com/datasheets/ThreatMetrix-SmartID-Datasheet.pdf
[TREND] Trend Micro, http://www.trendmicro.com/us/index.html
[VYAT] Vyatta, http://www.vyatta.com/
[WHIT] White Hat Security, https://www.whitehatsec.com/sentinel_services/sentinel_services.html
[ZSCA] Zscalar, http://en.wikipedia.org/wiki/Zscaler

第五部分总结

第三部分描述了云计算概念，第四部分描述了云计算实验系统，相应地，本部分描述了云计算安全概念、技术和产品。

第 16 章讨论了云计算安全的各种概念。首先，讨论了云计算安全和管理。其次，我们讨论了云的安全架构。再次，讨论了云的访问控制和身份管理。其后，讨论了云的数据安全问题。之后，我们概述了云的隐私、合规性和取证。最后，我们讨论了如加密解决方案、网络安全问题、业务连续性、操作安全性和云的物理安全性等方面。

第 17 章讨论了云计算功能安全的各个方面。我们概述了我们的云计算框架，然后讨论了云的虚拟化、网络、数据存储和管理的安全性。

第 18 章讨论了数据管理功能安全，然后描述了云对这些功能的影响。特别是，我们讨论了自主安全、对象安全、分布式数据管理安全和云安全的异构数据集成。

第 19 章概述了 NIST 所讨论的云计算安全指南。指南包括部署模型、Web 服务以及管理、合规性、可用性、身份和访问管理、体系结构、事件响应、信任、软件隔离和数据保护。

第 20 章讨论了安全即服务的概念，并以用于安全的数据挖掘服务为例进行了解释。这些数据挖掘服务是在云计算上实施的理想选择。首先讨论了用于网络安全的数据挖掘，然后简要介绍了我们正在开发的工具。

第 21 章概述了为云提供安全解决方案，或为各种业务提供安全服务的各种商业产品。其中一些产品主要用于云端或在云端运行。其他一些解决方案已经采用了一般安全产品并将其应用于云端。云安全解决方案包括加密、虚拟化安全和云监控。安全即服务包括身份管理、恶意软件检测以及网络和电子邮件过滤。

第六部分　云计算安全实验系统

第六部分简介

既然我们已经提供了关于云计算安全的概述，那么接下来将给出一些我们已经开发的云计算安全实验系统。这将使读者更好地了解我们是如何利用云系统中的各种访问控制模型的。第六部分由 3 章组成，即第 22 章、第 23 章和第 24 章。在第 22 章中讨论在云中的关系数据查询处理器安全。云中的语义 Web 数据查询处理器安全在第 23 章讨论。云中信息的安全集成将在第 24 章中进行讨论。

我们基于云的安全关系数据管理器利用 Hive 进行数据管理，并构建基于 XACML 的访问控制策略引擎。关于云中的语义 Web 数据处理安全，我们构建了一个 SPARQL 查询优化器，并开发了基于 XACML 的访问控制模型。最后，为了在云中实现安全的信息集成，我们建立了一个用于在 AMAZON S3 上存储数据的安全模块。我们开发的实验系统是文献中早期讨论的云数据安全管理系统之一。

第 22 章　对关系数据进行安全的云查询处理

22.1　概述

WWW 被设想为使用互联网（见本章参考文献［WWW］）访问的互联超文本文件系统。随着亚马逊（Amazon. com）和社交网络应用（如 Facebook 和 Twitter）等电子商务的出现，日常产生的数据量是巨大的（见本章参考文献［AXON10］）。据估计，在 2009 年由个人产生的数据量将超过 2008 年人类全年产生的数据量（见本章参考文献［WEIG09］）。一个组织所产生的大量数据，如果和其他组织或研究人员所拥有的数据相关联的话，那么这些数据对于后者来说可能是有价值的。对于各种政府情报机构更是如此。这导致一种趋势：商业组织和大学之间建立了伙伴关系进行合作研究（见本章参考文献［NOKI］）；商业组织之间也建立伙伴关系，进行数据共享去创造更好的应用程序（见本章参考文献［SALE08］）。

在组织之间进行这种合作过程，有两个主要障碍：一是安排一个大型、通用的数据存储区域，二是提供对共享数据的安全访问。全球各地的企业投入大量资源，以尽量减少存储成本，并引入云计算服务，这样成本的预算将进一步降低（见本章参考文献［ALL10］）。此外，尽管各个组织在安全性方面支出大量的年度预算，但这还不足以防止安全漏洞（见本章参考文献［SAWY10，FIRS10］）。在本章中，我们提出了一个基于 Web 的系统（Hive 访问控制），旨在通过将云计算技术与基于策略的安全机制相结合来实现上述目标。这个想法部分来自于身份与访问管理云安全联盟（见本章参考文献［KUMA10］）和我们以前使用 XACML 策略的工作（见本章参考文献［PARO09］）建议。我们将 HDFS（见本章参考文献［BORT10］）与 Hive（见本章参考文献［APACa］）相结合，为参与该项目的组织提供了一个共同的存储区域。除此之外，我们已经使用了基于策略的 XACML（见本章参考文献［MOSE05］）安全机制来对共享数据提供细粒度的访问控制。我们的系统根据可以运行的查询类型（如 SELECT 和 INSERT）将用户分为几组，并基于加盐哈希技术为用户提供安全的登录功能。当用户登录到我们的系统时，根据用户所属的组，他/她被提供不同的选项。我们允许协作组织以关系表和视图的形式，将数据加载到共享存储空间。用户才可以为用户组的表/视图定义细粒度的 XACML 访问控制策略。用户还可以根据所拥有的凭证来查询整个数据库。我们在系统中提供了一些基本的查询重写规则，从 Hive（HiveQL）的查询语言中抽象出用户。这允许他们在 Web 应用程序中输入常规的 SQL 查询，这个查询是使用基本重写规则转换的 HiveQL。我们的系统还允许新用户注册，但只有指定的特殊“管理员”用户才可以将这些用户分配给适当的组。我们做出的贡献包括：

■ 使用 Hive 存储在 HDFS 中的安全加载和查询共享数据的机制。

■ 在 HDFS 和 Hive 之外的其他安全层使用基于 XACML 策略的机制。

■ 基本的查询重写规则，用于从 HiveQL 中抽象出用户，并允许他/她输入 SQL 查询。

■ 将上述机制并入基于网络的系统。

本章的结构如下：22.2 节介绍云中信息安全存储和检索领域的相关工作。在 22.3 节中，介绍我们的架构，用于解决基于云计算技术和基于 XACML 策略的安全机制的大规模数据安全共享问题。此外，在 22.4 节中，我们将介绍我们的实施细节。最后，22.5 节介绍我们的结论和未来的工作。图 22.1 说明了本章的内容。

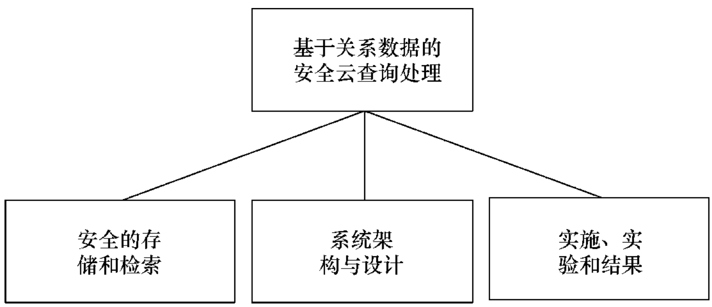


图 22.1　使用关系数据进行安全云查询处理（源自 Thuraisingham, B.，M. V. Khadilkar，A. Gupta，M. Kantarcioglu 和 L. Khan，Secure data storage and retrieval in the cloud. CollaborateCom，Chicago，IL，1－8，2010.）

22.2　相关工作

我们将云计算技术与安全机制相结合，使合作组织能够安全地共享大量数据。自云计算技术诞生以来，研究人员、商业组织和媒体对这些技术的安全问题均产生了浓厚的兴趣（见本章参考文献［TALB09］、［MITC096］）。这种兴趣导致了商业组织（见本章参考文献［WIND］、［AMAZ09］、［OMAL09］）的大规模研发工作。与云计算安全相关的一部分工作是一直侧重在基础架构层面实施安全性。在本章参考文献［OMAL09］中，作者介绍了他们在 Hadoop 中的安全观。从而在 Hadoop 发现了一些安全隐患，并为此提供了解决方案。这些解决方案已经在 Hadoop v0.20 的 beta 版本中实现。这一开发工作是保护云基础设施的重要一步，但仅在其初级阶段。我们系统的目标是在 Hadoop 提供的安全性之上添加另一层安全性。一旦 Hadoop 提供的安全性变得强大，它会增强我们系统的有效性。

AWS 是云中的一种网络服务基础设施平台（见本章参考文献［AMAZ］）。本章参考文献［AMAZ09］提供了与 AWS 相关的安全方面的概述，例如物理安全性、网络安全性和 AWS 安全性。我们的系统与 AWS 不同，因为我们的云基础设施完全是私有的，

而 AWS 的基础设施处在公有领域。这个相互区别的因素使得我们的基础设施“信任度”超过 AWS 基础架构，其中 AWS 的数据必须以加密格式存储，因为它是公有领域的。未来，我们打算扩大我们的工作范围，包括公有云和私有云。

Windows Azure 平台是在 Microsoft 数据中心（见本章参考文献［WIND］）中托管的互联网规模的云计算服务平台。本章参考文献［MARS10］概述了为 Windows Azure 平台设计和开发更安全应用程序的安全挑战和推荐方法。然而，根据本章参考文献［BROD10］，Windows Azure 平台适用于构建新应用程序，移植现有应用程序并不是最佳选择。我们没有使用 Windows Azure 平台的主要原因，是希望将现有的应用程序移植到开源系统，而不是像 Windows Azure 那样从头开始编写代码。我们也不想绑定到 Windows 框架，而是想允许我们的工作在任何类型的系统上都可以使用。一旦平台支持使用虚拟机运行现有应用程序（见本章参考文献［BROD10］），我们就可以在 Windows Azure平台上测试我们的系统。

22.3 系统架构

在本节中将介绍我们的架构，它可以安全地访问大型公共存储空间（“云”），从而允许合作组织间可靠地共享数据。我们首先对架构进行概述，然后讨论其每个组件层。

图 22.2 显示了系统的架构。图中的每个矩形代表了框架的不同组成部分。箭头的各种线条样式，表示可以使用此系统完成特定任务的控制流程。接下来，我们介绍架构中的每个组件层。

22.3.1 Web 应用层

Web 应用层是系统向用户提供访问云基础架构的唯一接口。我们根据分配给用户的权限提供不同的功能。Web 应用程序提供可由任何用户登录系统的登录页面。我们使用 Java 简化加密（Java – Simplified encryption，简称为 JASYPT）库的（见本章参考文献［JASY］）加盐哈希技术，将用户名和密码存储在文件中。此外，该文件存储在任何用户无法访问的安全位置。目前，系统支持 3 种类型的用户：

■ 只能查询现有表/视图的用户；

■ 除了查询所有表/视图之外，还可以创建表/视图并在其上定义 XACML 策略的用户；

■ 除了以上的功能之外，一个特殊的“管理员”用户还可以将新用户分配给上述类型用户。

22.3.2 ZQL 解析器层

ZQL 解析器（见本章参考文献［ZQL］）层使用用户提交的任何查询作为输入，不管查询被解析成功还是向用户返回错误消息，都会继续执行 XACML 策略评估程序。ZQL 解析器是一个用 Java 编写的 SQL 解析器，它将 SQL 查询作为输入，并用查询（见

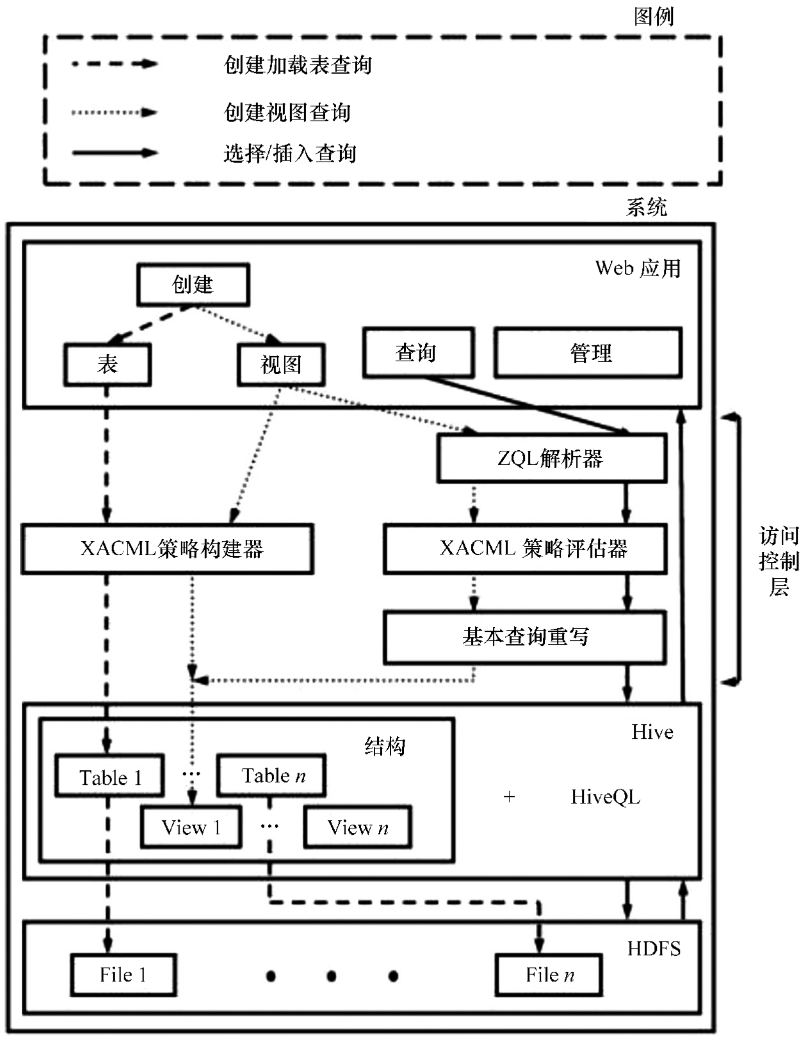


图 22.2　系统架构

本章参考文献［ZQL］）的不同部分填充不同的 Java 向量。例如，以下查询：

SELECT a.id, a.name FROM a WHERE a.id > 5.

ZQL 解析器解析查询，并为查询的每个部分（SELECT、FROM 和 WHERE）构造不同的 Java 向量。在我们的系统中，上述查询的 SELECT 子句中的属性名称向量将返回到 Web 应用层，用于显示查询返回的结果。FROM 子句中的表/视图名称向量传递给 XACML策略计算器，以确保当前用户具有访问查询中指定的所有表/视图的权限。如果评估者确定当前用户具有所需的权限，则会进一步处理查询，否则将向 Web 应用层返回错误消息。目前，ZQL 解析器支持 DELETE、INSERT、SELECT 和 UPDATESQL 语句。我们未来的工作涉及添加对其他关键字（如 CREATE、DROP 等）的支持。

22.3.3 XACML 策略层

XACML 是一种基于 XML 的语言，用于定义资源访问控制策略。并且使用相同的语言来确定是否允许访问基于策略定义的特定资源（见本章参考文献［ZQL］）。接下来，我们解释在框架中如何定义和使用 XACML 策略。

22.3.3.1 XACML 策略构建器

在我们的框架中，用户定义的表和视图被视为构建 XACML 策略的资源。此外，针对系统提供的多种查询涉及的资源，定义了基于角色的访问控制（见本章参考文献［FERR92］、［SAND96］）策略。对于我们框架所支持每种类型的查询，定义了此类型与允许运行该类型查询的所有用户之间的映射。以下给出了此类映射的示例列表：

INSERT admin user1 user2

SELECT admin user1 user3

在我们的系统中，用户想要创建的每个表/视图都可以选择上传自己预定义的 XACML 策略，或让框架为它们构建一个策略。如果用户选择后一个选项，它们还必须指定表/视图上允许的查询种类（例如 INSERT、SELECT 等）。然后，我们使用 Sun 的 XACML 实现（见本章参考文献［SUN］）去为该表/视图构建由特定用户来指定组的策略。

22.3.3.2 XACML 策略评估器

我们的系统使用 Sun 的 XACML 实现（见本章参考文献［SUN］），来评估当前用户是否可以访问在任何用户查询中定义的所有表/视图。如果授予所有表/视图的查询权限，则可进一步处理查询，否则将向用户返回错误消息。策略评估器在常规用户查询执行期间，以及在创建视图期间都使用，因为在 Hive 中创建视图的唯一方法，是通过在现有表/视图上指定 SELECT 查询。在创建视图之前，当前用户必须能够访问此 SELECT 查询中指定的所有表/视图。

22.3.3.3 基本查询重写层

该层使我们能够在用户和 HiveQL 之间添加另一层抽象层，允许用户根据 HiveQL 语法输入重写的 SQL 查询。在我们目前的系统中，为用户指定的 SQL 查询提供了两个基本的重写规则。

HiveQL 不允许查询中的 FROM 子句中有多个表，而可以将这种查询作为一系列 JOIN 语句来给出。通过允许用户对多个表使用常规 SQL 查询，从而将用户从事实中抽象出来，这多个表来自 FROM 子句，FROM 子句也将转换为符合 HiveQL 语法的 JOIN 语句序列。以下是一个例子：

SELECT a.id, b.age FROM a, b; → SELECT a.id, b.age FROM a JOIN b;

首先，HiveQL 使用 SQL 的 INSERT – SELECT 语句的修改版本，INSERT OVERWRITE TABLE <表名称> SELECT，而不是 INSERT INTO <表名称> SELECT。其次，我们通过允许他/她输入传统的 INSERT INTO <表名称> SELECT。最后，将其重写为 HiveQL 的 INSERT OVERWRITE TABLE <表名称> SELECT，去把用户抽象出来。以下是一个例子：

INSERT INTO a SELECT * FROM b; → INSERT OVERWRITE TABLE a SELECT * FROM b.

作为我们未来工作的一部分，我们计划用完整的查询重写引擎中更复杂的规则，去扩展这些基本的重写规则。

22.3.3.4　Hive 层

Hive 是建立在 Hadoop 之上的数据仓库基础架构。Hive 提供了在基础 HDFS 中构建数据的能力以及查询这些数据的功能。图 22.2 中，这一层的表格与 HDFS 图层中文件之间的箭头，表示 Hive 中的每个表格都作为文件存储在 HDFS 中。这些文件包含此表所代表的数据。由于视图只是使用 SELECT 查询创建的 Hive 中的逻辑概念，因此层中的视图和 HDFS 层中的文件之间没有箭头。在我们的框架中，Hive 用于构建将由合作组织共享的数据。此外，我们使用 Hive SQL 查询语言来访问这些数据。在我们的系统中使用 Hive 的优点是用户可以使用熟悉的类似 SQL 的语法来查询数据。

22.3.3.5　HDFS 层

HDFS 是一种分布式文件系统，被设计用于在基本硬件（见本章参考文献［BORT10］）上运行。我们框架中的 HDFS 层存储与 Hive（见本章参考文献［THUS09］）中创建的表相对应的数据文件。我们的安全假设是这些文件既不能使用 Hadoop 的（见本章参考文献［APACb］）Web 界面，也不能使用 Hadoop 的命令行接口（Command Line Interface，CLI），只能使用我们的系统。

22.4　实验细节和结果

在本节中，我们通过提供不同大小表的插入和查询过程的性能图，来介绍我们系统的实现。我们的实验和性能图的细节在本章参考文献［THUR10］中给出。下面先简要介绍一下我们的实验配置，然后是实现细节。

22.4.1　实验配置

我们的实验是在一个 19 节点集群上进行的，并且具有两种不同配置的节点混合。此外，所有节点都在同一个机架中。在 19 个节点中，11 个节点在具有 4 GB SDRAM 400 MHz 内存的 Intel Pentium 4，3.2GHz 中央处理器（CPU）和作为主驱动器的 40GB Western Digital WDC WD400BB－75FJ SATA 硬盘驱动器以及作为辅助驱动器的 250GB Western Digital WD2500AAJB－0 SATA 硬盘上运行 Ubuntu v10.04 Lucid Lynx，另外 8 个节点在具有 4 GB SDRAM 333 MHz 内存的 Intel Pentium 4 2.8 GHz CPU 和两个 40 GB Western Digital WDC WD400BB－75FJ SATA 硬盘驱动器上运行 Ubuntu v9.04 Jaunty Jackalope。我们使用 Java 版本 JRE v1.6.0 \ _ 18 来实现。对于云基础架构，我们使用 Hadoop v0.19.1 版本（其具有 1000MB 堆空间）以及 Hive 版本 v0.5（其还具有默认堆空间）。我们还使用 Apache Tomcatv7.0.0 作为应用程序的 Web 服务器，具有 2 GB 的堆空间。我还为 Hadoop 和 Hive 提供的所有参数使用了默认值。我们了解到，当设置这些参数为最

佳值时，将获得性能的提升。但由于这是一项初步工作，我们现在不用重点关注这些参数，未来将要进一步完善。

22.4.2 实验数据集

我们使用两个不同的数据集来测试我们的系统与 Hive 的性能。第一个数据集是 Freebase（见本章参考文献［FREE］）系统，它是一个开放的结构化数据库，拥有约 1200 万个主题或实体。实体是具有唯一标识符的人、地点或事物。我们想通过使用 Freebase 系统中的人员、业务、电影、体育、组织和奖励数据集来模拟合作组织的环境。我们假设每个数据集由不同的组织加载到系统中，而且还可以基于用户的权限在这些数据集之间执行各种查询。用于测试实验的查询是由基于 Freebase 系统的数据集创建的。

用来测试系统的第二个数据集是著名的 TPC－H 基准测试（见本章参考文献［TPC］)。TPC－H 基准是一个决策支持的基准，由一个对任何商业组织来说都比较典型的模式组成。该基准测试包含 8 个表，并提供 22 个具有高度复杂的查询。我们使用这个基准，测试我们的系统与 Hive 在执行复杂查询时的性能。TPC－H 基准测试提供了数据生成工具（DBGEN）和查询生成工具（QGEN)。我们已经使用 DBGEN 生成基准文件中规定的具有 1～1000 的不同比例因子（SF）的数据集。读者应注意，比例因子为 1 是大约 1GB 的数据。因此，我们对系统的测试，数据大小从 1～1000GB 不等。较小的数据集（SF ＝ 1，SF ＝ 10 和 SF ＝ 30）用于测试我们的系统与 Hive 的加载性能。另外，较大的数据集（SF ＝ 100，SF ＝ 300 和 SF ＝ 1000）用于运行几个基准查询。使用 TPC－H 基准测试的 Q1、Q3、Q6 和 Q13 来测试我们的系统。原始查询不需要手动分为子查询，因为我们的 Web 应用程序不支持此功能。我们还认为，通过运行上述查询获得的结果，可以展示出在我们的系统上运行所有其他 TPC－H 基准查询的性能。

22.4.3 实验结果

我们已经对基于 Web 的系统进行了测试，例如数据加载和查询时间等性能指标。此外，我们已经将这些指标与 Hive CLI 进行了比较。在本小节的性能图表和结果表中使用的所有查询时间均通过三次单独运行后取平均。我们运行了两组实验，一组使用 Freebase 系统，另一组使用 TPC－H 基准测试。

我们已经比较了为 Freebase 数据集加载和查询应用程序数据所需的时间与 Hive 的时间。应用程序的数据加载时间几乎与 Hive 的小型表（0.1 万和 0.3 万元组）的时间相同。随着元组的数量增加到 50 万，然后增加到 167 万，我们的系统加载元组比 Hive 慢。这主要是因为建立与 Hive 数据库连接相关联的开销，以及与正在加载的表构建 XACML 策略需要相应的时间。

此外，我们比较了应用程序和 Hive 之间的简单“SELECT ＊ FROM”查询的运行时间。已经在用于数据加载的相同表上执行查询，但是通过仅使用前 100 个元组的 LIMIT 子句来限制我们的结果。这样做是为了避免在我们的应用程序和 Hives CLI 之间出现较大的时间差异，因为我们已经在结果上实现了分页机制，而 Hive 的 CLI 将在屏幕上显

示所有结果。我们的应用程序运行时间比 Hive CLI 上查询的运行时间稍短。这个差异是由于 Hive CLI 在屏幕上显示查询结果所花费的时间。两个运行都很快，因为 Hive 不需要为此查询运行 Map/Reduce（见本章参考文献［APACc］）作业，而只需要从 HDFS 返回相应表的整个文件。

我们还运行了许多其他查询，并将我们系统上的这些查询的运行时间与 Freebase 系统的 Hive 进行了比较。这些查询能测试我们的系统在创建和查询视图/表时与 Hive 的性能。我们测试了各种查询，包括插入、创建、选择、聚合和连接查询。

我们还比较了 TPC－H 基准中 SF ＝ 1、10 和 30 的“客户”和“提供商”表的应用与 Hive 的数据加载时间。目前，我们的系统允许用户上传最多 1 GB 的数据文件。TPC－H 基准测试的 DBGEN 工具为 SF ＝ 1、10 和 30 的“客户”和“提供商”表生成小于 1 GB 的文件。这也是为什么选择 SF ＝ 1、10 和 30 的“客户”和“提供商”表，来比较我们系统与 Hive 的数据加载性能的原因。在数据加载方面，我们的系统在最小 SF 为 1 时执行与 Hive 相似，随着 SF 的增加，我们的系统比 Hive 慢。再次声明，执行性能的差异是由于与 Hive 数据库连接和 XACML 策略生成需要相应的开销。我们的系统和 Hive 共同的趋势是预期的线性关系，因为这些表的大小随着 SF 线性增加。

最后，比较了我们系统上的 4 个 TPC－H 基准查询与 Hive 的性能。我们的系统和所选查询的 Hive CLI 一样。在两者的查询性能时间的基础上，随着表格大小的增加，基准查询执行的时间也如预期的那样增加。在生产环境中，这些查询将不会在运行时针对大型数据集执行。我们宁愿离线运行这些查询，并可以将结果作为视图存储在 Hive 中。然后，我们也可以使用查询重写机制，来有效地返回这些结果。

22.5 总结和展望

在本章中，我们提出了一个允许各合作组织安全地共享大量数据的系统。我们通过使用 Hadoop 来确保各个组织有一个很大的共同存储区域。此外，我们已经使用 Hive 可以向系统的用户展示数据的结构化视图，并且还可以使用类似 SQL 的语言来查询数据。我们使用一个简单的加盐哈希机制，来验证当前版本系统中的用户。我们计划在未来版本的系统中实施更复杂的身份验证技术。在本章中，我们使用 ZQL 解析器，来解析用户输入的任何 SQL 查询。我们还计划扩展此解析器，并支持目前在 ZQL 中不支持的如 DESCRIBE、JOIN 等关键字。我们通过实现一些基本的查询重写规则，将用户从使用 Hive 中抽象出来。我们未来工作的一部分是在 Hive 中实现物化视图，并将基本的查询重写规则扩展到一个完整的 Hive 引擎，该引擎考虑到所有的表/物化视图和它们定义的 XACML 策略。我们使用 XACML 策略为共享数据提供了细粒度的访问控制，还根据用户提交给我们系统的查询种类，在我们的框架中引入基于角色的访问控制。在我们系统的当前版本中，我们只提供两种类型的关键字，INSERT 和 SELECT 作为 XACML 策略组的支持。

在将来，我们计划将系统扩展到其他基于关键字的组，如 DELETE、UPDATE 等。我们还计划使用 Hadoop 和 Hive 提供的参数，来测试对查询执行性能的不同值的影响。最后，目前的系统只在私有云中实现，将在未来的版本中，要扩展到包括公有云在内的

云中，如 AWS 和 Amazon 简单存储服务。

参 考 文 献

[ALL10] All, A., Cloud data storage: It'll still cost you, so give it some thought, http://www.itbusinessedge.com/cm/blogs/all/cloud-data-storage-itll-still-cost-you-so-give-it-some-thought/?cs=38733, January 2010.

[AMAZ] Amazon Web Services, http://aws.amazon.com/.

[AMAZ09] Amazon Web Services: Overview of security processes, http://awsmedia.s3.amazonaws.com/pdf/AWS_Security_Whitepaper.pdf, November 2009.

[APACa] Apache Hive, https://cwiki.apache.org/confluence/display/Hive/Home%3bjsessionid=8946C5F66E7FBD0CE2466BAA5C699289.

[APACb] Apache Hadoop, http://hadoop.apache.org/.

[APACc] Apache MapReduce, http://hadoop.apache.org/mapreduce/.

[AXON10] Axon, S., Facebook will celebrate 500 million users next week, http://mashable.com/2010/07/17/facebook-500-million/, July 2010.

[BORT10] Borthakur, D., HDFS architecture, http://hadoop.apache.org/common/docs/r0.19.2/hdfs_design.pdf, 2010.

[BROD10] Brodkin, J., Microsoft Windows Azure and Amazon EC2 on collision course, http://www.networkworld.com/news/2010/062510-microsoft-azure-amazon-ec2.html, June 2010.

[FERR92] Ferraiolo, D. F. and D. R. Kuhn, Role-based access controls. In *National Computer Security Conference*, pp. 554–563, Baltimore, MD, 1992.

[FIRS10] First annual cost of cyber crime security, Technical report, Ponemon Institute, July 2010.

[FREE] Freebase, http://www.freebase.com/.

[JASY] JASYPT—Java simplified encryption, http://www.jasypt.org/index.html.

[KUMA10] Kumaraswamy, S., S. Lakshminarayanan, M. Reiter, J. Stein, and Y. Wilson, Domain 12: Guidance for Identity & Access Management V2.1, April 2010.

[MARS10] Marshall, A., M. Howard, G. Bugher, and B. Harden, Security best practices for developing Windows Azure applications, June 2010, http://download.microsoft.com/download/7/3/E/73E4EE93–559F-4D0F-A6FC-7FEC5F1542D1/SecurityBestPracticesWindowsAzureApps.docx

[MITC096] Mitchell, R. L., Cloud storage triggers security worries, http://www.computerworld.com/s/article/340438/Confidence in the Cloud?, July 2009.

[MOSE05] Moses, T., eXtensible access control markup language (XACML) Version 2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, February 2005.

[NOKI] Nokia Research Center—Open innovation, http://research.nokia.com/openinnovation.

[OMAL09] O'Malley, O., K. Zhang, S. Radia, R. Marti, and C. Harrell, Hadoop security design, http://bit.ly/75011o, October 2009.

[SALE08] Salesforce.com and Facebook create new opportunities for enterprise applications to serve Facebook's 120 million users, http://www.salesforce.com/company/news-press/press-releases/2008/11/081103-4.jsp, November 2008.

[SAND96] Sandhu, R. S., E. J. Coyne, H. L. Feinstein, and C. E. Youman, Role-based access

control models. *IEEE Computer*, 29(2): 38–47, 1996.

[SAWY10] Sawyer, J., Tech insight: How to cut security costs without a lot of pain, http://www.darkreading.com/smb-security/security/management/showArticle.jhtml?articleID=226200159, July 2010.

[SUN] Sun XACML implementation, http://sunxacml.sourceforge.net/.

[TALB09] Talbot, D., How secure is cloud computing? http://www.technologyreview.com/computing/23951/, November 2009.

[THUR10] Thuraisingham, B., M. V. Khadilkar, A. Gupta, M. Kantarcioglu, and L. Khan, Secure data storage and retrieval in the cloud. *CollaborateCom*, Chicago, IL, 1–8, 2010.

[THUS09] Thusoo, A., J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, Hive—A warehousing solution over a map-reduce framework. *PVLDB*, 2(2): 1626–1629, 2009.

[TPC] TPC BENCHMARK H, http://tpc.org/tpch/spec/tpch2.11.0.pdf.

[WEIG09] Weigend, A., The social data revolution(s), http://blogs.hbr.org/now-new-next/2009/05/the-social-data-revolution.html, May 2009.

[WIND] Windows Azure platform—Whitepapers, http://www.windowsazure.com/en-us/develop/net/other-resources/white-papers/

[WWW] World Wide Web, http://en.wikipedia.org/wiki/World Wide Web.

[ZQL] Zql: A Java SQL parser, http://zql.sourceforge.net/.

第 23 章　对语义 Web 数据的云查询处理安全

23.1　概述

语义 Web 变得越来越普遍。大多数小型和大型企业（如 Oracle、IBM、Adobe、Software AG 等）正在积极使用语义 Web 技术（见本章参考文献［W3C09］），而且在更广泛的应用领域如医疗保健和生命科学领域正在考虑将其应用于数据集成的可能性（见本章参考文献［W3C09］）。Tim Berners－Lee 爵士最初设想了语义 Web 作为可被机器理解的网络（见本章参考文献［LEE98］）。语义 Web 的力量在于其对网络资源之间关系的整合（见本章参考文献［W3C09］）。

语义 Web 连同本体一起，是表示知识的最强大的方法之一。本体正式地描述域中的概念或类、类的各种属性、类之间的关系和限制。知识库可以由本体及其各种类实例构建。知识库（本体及其实例）的一个例子如图 23.1 所示。

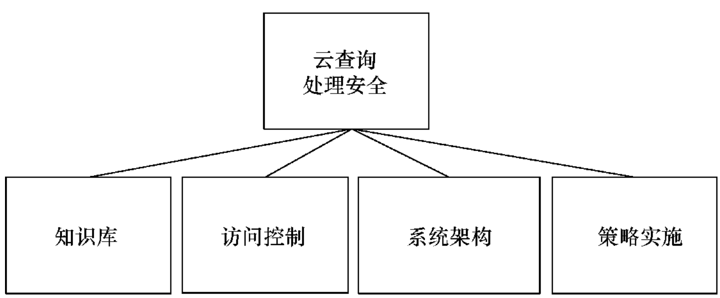


图 23.1　云查询处理安全

RDF 由于其表现力、语义互操作性和可重用性而被广泛用于语义 Web。目前使用的大多是 RDF 存储，其中包括 Apache Jena Fuseki（见本章参考文献［APAC］）、Kowari（见本章参考文献［KOWA］）、3store（见本章参考文献［HARR05］）和 Sesame（见本章参考文献［BROE02］），但是它们并不在意其安全性。我们已经努力去考虑安全性，特别是在 Jena 中（见本章参考文献［JAIN06］、［REDD05］）。然而，Jena 的一个缺点是缺乏可扩展性。因此，较大数据集的执行时间可能变得相当缓慢，使得大型存储的某些查询难以处理（例如，具有 1000 万次或更多次的数据）（见本章参考文献［HUSA09］、［HUSA10］）。

另一方面，由于可扩展性、并行处理能力、成本效益和可用性，大型 RDF 可以在云计算机中存储和检索。最广泛使用的云计算环境 Hadoop（Apache）（见本章参考文献［APAC］）使用了 Google 的 Map/Reduce 框架。Map/Reduce 将大型作业分解为较小的作

业，并将这些作业的结果合并，以便在完成子任务后，生成最终的输出。先前的工作表明，可以在这些云中高效地存储和查询大型 RDF 图（见本章参考文献［CHOI09］、［HUSA09］、［HUSA10］和［MIKA08］）。在存储和管理的同时，大型 RDF 图已经受到关注，而 Hadoop 中 RDF 存储的访问控制很少受到关注。在本章中，我们将描述一种在 Hadoop 上实现 RDF 数据访问控制的系统。

我们的系统实现了基于令牌的访问控制系统。系统管理员根据代理的需求和安全级别，为安全性相关数据授予访问令牌（Access Token，AT）。由于将相互冲突的访问令牌分配给同一代理机构可能会产生冲突，这可以通过使用访问令牌的时间戳来解决。对此，我们使用 LUBM（见本章参考文献［GUO05］）测试实例进行实验。另外，Hadoop 中已经生成和实现了一些示例的场景。

我们已经做出了一些贡献。首先，我们设计了一个扩展到极大数据集的架构；第二，我们不仅在用户层面，还在主体、客体和谓词的层次上处理访问控制，使策略具有细粒度，比以前的产品更具表现力；第三，我们设计了基于时间戳的冲突检测和解决算法；第四，对架构进行了实验，并选择了以下几个阶段测试了基准数据：查询重写（预处理阶段）、嵌入式执行（Map/Reduce 执行阶段）和后处理执行（数据显示阶段）。最后，整个系统在 Hadoop——一个开源云计算环境中实现。本章对于 Hadoop 中的 RDF 数据所考虑的其他访问控制是有好处的。

本章的结构如下。在 23.2 节中，我们将介绍 Hadoop 和 Map/Reduce 的相关工作和简要概述。23.3 节介绍访问令牌、AT 元组、冲突和我们对冲突的解决算法。在 23.4 节中描述我们系统的架构。在 23.5 节中，描述将 AT 分配给代理的影响，包括实验及其运行时间。最后，23.6 节将阐述本章总结和未来的工作建议。图 23.2 说明了本章包含的内容。

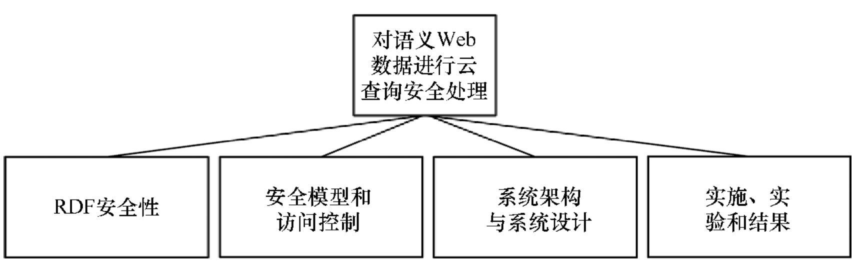


图 23.2　对语义 Web 数据进行云查询安全处理［源自 Khaled，A.，Husain，M. F.，Khan，L.，Hamlen，K. W. A Token - Based Access Control System for RDF Data in the Clouds，*Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science* （CloudCom），p. 104 - 111，2010. ©（2010）IEEE.］

23.2　背景

23.2.1　相关工作

我们首先描述以前关于单机 RDF 安全性的工作。然后，总结存储 RDF 数据的一些

云计算架构。最后，介绍我们自己以前的工作总结。

虽然人们对 RDF 知识的存储、表示和推理进行了广泛的研究，但对于 RDF 存储的安全性和访问控制问题的研究相对较少（见本章参考文献［REDD05］）。Reddivari 等人（见本章参考文献［REDD05］）已经实施了基于一组规则策略的访问控制。它们处理 RDF 存储中的三元组、模型和集合的插入/删除操作，以及查看和使用操作。Jain 和 Farkas（见本章参考文献［JAIN06］）将 RDF 保护对象描述为 RDF 模式，并为其设计了符合安全性的要求。他们表明了子类或子属性的安全级别，应该至少与超类型一样受限制。本章参考文献［KIM08］中讨论的基于 RDF 三重访问控制模型，考虑了显式和隐式的授权传播。

以上这些工作大部分都是在 Jena 中实现的。然而，Jena 的扩展性差，因为它运行在单台机器上，无法处理大量的数据（见本章参考文献［HUSA09］、［HUSA10］）。Husain 等人（见本章参考文献［HUSA09］、［HUSA10］）设计和实现一个架构来存储和查询大型 RDF 图。Mika 和 Tummarello（见本章参考文献［MIKA08］）将 RDF 数据存储在 Hadoop 中。SPIDER 系统（见本章参考文献［CHOI09］）存储和处理大型 RDF 数据集，但缺少访问控制机制。

我们的架构包含一个访问控制层来支持大型数据集的访问控制，这个架构在本章参考文献［HUSA10］中讨论。我们的方法不是将访问控制直接分配给用户或代理，而是为特定访问级别生成令牌，并根据业务需求和代理的安全级别，将令牌分配给代理。虽然其他人使用令牌进行访问控制来管理 XML 文档（见本章参考文献［BOUG04］）和数字信息（见本章参考文献［HOLM99］），但这些标记尚未用于 RDF 存储。使用令牌的一个优点是，如果多个代理的需求和安全要求相同，它们就可以重复使用。

23.2.1.1 Hadoop 和 Map/Reduce

接下来，我们简要介绍 Hadoop（见本章参考文献［APAC］）和 Map/Reduce。在 Hadoop 中，计算单位称为作业。用户向 Hadoop 的 JobTracker 组件提交作业。每个工作有两个阶段：映射（Map）和规约（Reduce）。映射阶段将键值对作为输入，并可输出零个或多个键值对。在 Reduce 阶段，每个键的值都被分组到一起构成集合，由迭代器遍历。然后将这些迭代键值对器传递给 Reduce 方法，Reduce 方法再输出零个或多个键值对。当作业提交给 JobTracker 时，Hadoop 会尝试将 Map（映射）进程靠近集群中的输入数据。每个 Map 过程和 Reduce 过程都可以独立运行而无需通信。这种无通信方式，有利于速度的提升和实现的简单性。

23.3 访问控制

23.3.1 模型

定义 23.1

AT（访问令牌）允许访问与安全相关的数据。持有 AT 的代理可以查看该 AT 所允

许的数据。我们用正整数表示 AT。

定义 23. 2

访问令牌元组（ATT）具有 <访问令牌、元素、元素类型和元素名称>的形式，其中元素可以是主体、客体或谓词，元素类型可以被描述为 URI、DataType、Literal、Model 或 BlankNode。模型用于访问主体模型，稍后将在本节中进行说明。

例如，David 是一个主体，<1，Subject，URI，David>是一个 ATT。具有 AT1 的任何代理可以检索所有文件上的 David 的信息（遵守与 David 对象相关联的 URI、文字等访问的任何其他安全限制）。在描述 icates 的 ATT 时，我们将元素名称留空（_ ）。

在记录组织的基础上，我们支持 6 个访问级别，以及下面描述的几个对应的子类型。可以为代理分配一个或多个以下访问级别。这里，把具有共同 AT 的访问级别结合在一起，而具有不同 AT 的访问级别分开组合。

1）谓词数据访问：如果对访问级别中的一个特定谓词定义了对象类型，则具有该访问级别的代理可能会读取整个谓词文件（受任何其他策略限制）。例如，<1，谓词，isPaid，_ >是一个 ATT，它允许其拥有者读取整个谓词文件 isPaid。

2）谓词和主体数据访问：拥有主体 ATT 的代理，可以访问与特定主体相关联的数据，其中主体可以是 URI 或 DataType。将这些主体 ATT 中的一个与谓词数据访问相结合，具有相同 AT 的 ATT 授权代理，可以访问特定谓词的特定主体。例如：

① 谓词和主体作为 URI：组合 ATT 的<1，谓词，isPaid>和<1，主体，URI，MichaelScott>允许具有 AT1 的代理，访问具有 URI 的主体 MichaelScott 的谓词 isPaid。

② 谓词和主体作为数据类型：类似地，谓词和数据类型的 ATT 可以组合成主体，以允许通过特定谓词文件访问特定数据类型。

为了简洁起见，我们省略了对每个以下访问级别的不同主体和客体变体的描述。

3）谓词和客体：此访问级别允许主体提取满足特定谓词和客体的名称。例如，ATT 的<1，谓词，有维生素，_ >和<1，客体，URI，E>，具有 AT1 的代理可以查看具有维生素 E 的受试者（例如食物）的名称。更一般地，如果 X1 和 X2 是描述 AT 的谓词和客体三元组（分别）生成的三元组，则拥有 AT 的代理可以查看三元组的集合 $X1 \cap X2$。此示例如图 23. 3 所示。ATT 的联合需要与通用的 AT 一起组合。

4）主体访问权限：使用此访问级别，代理可以读取所有文件上的主体信息。这是限制较少的访问级别之一。主体可以是 DataType 或 BlankNode。

5）客体访问：使用此访问级别，代理可以通过所有文件读取客体的主体。像上一级一样，这是限制较少的访问级别之一。该客体可以是 URI、DataType、Literal 或 BlankNode。

6）主体模型级访问：模型级访问允许代理读取所有必需的谓词文件，以获取给定主体的所有客体。在这些客体中，作为 URI 的那些客户接下来被视为主体，来提取它们各自的谓词和客体。该过程继续迭代，直到所有客体最终都成为文字或空白节点。以这种方式，具有模型级访问的代理可以在给定主体上生成模型。

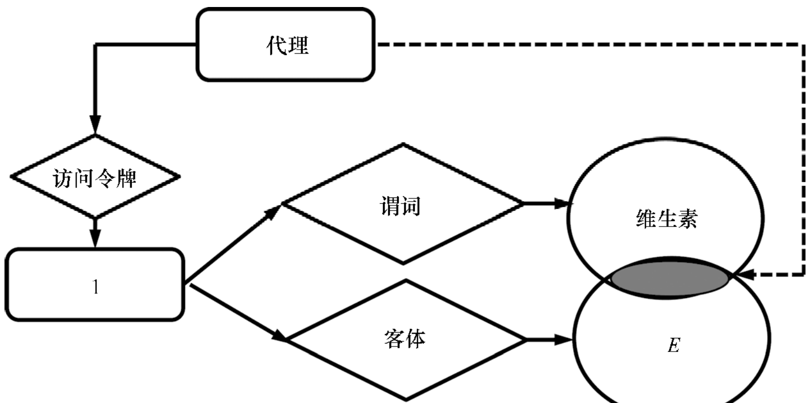


图 23.3 ATT 和通用 AT 的结合 [源自 Khaled，A.，Husain，M. F.，Khan，L.，Hamlen，K. W. A Token – Based Access Control System for RDF Data in the Clouds，*Proceedings of the* 2010 *IEEE Second International Conference on Cloud Computing Technology and Science*（CloudCom），p. 104 – 111，2010. ©（2010）IEEE.]

23.3.2 AT 任务

定义 23.3

访问令牌列表（AT – list）是授予给定代理的一个或多个 AT 的阵列，以及标识每个 AT 被授予的时间戳。要为每个代理维护单独的 AT 列表。

当系统管理员决定将 AT 添加到代理的 AT 列表时，AT 和时间戳首先被存储在临时变量 AT 中。在进行更改之前，系统必须首先检测新 AT 列表中是否存在潜在冲突。

23.3.2.1 代理 AT 的最终输出

每个 AT 允许访问一个三元组集合，我们将此集合称为 AT 的结果集。由代理人访问的三元组集合是代理人 AT 列表中 AT 的结果集合。也就是说，如果 Y_1，Y_2，…，Y_n 是 AT 的结果集，AT_1，AT_2，…，AT_n（分别）在代理的 AT 列表中，则代理可以访问集合 $Y_1 \cup Y_2 \cup \cdots \cup Y_n$ 中的三元组。

23.3.2.2 安全级别默认值

通过保守地选择系统中数据的默认安全级别，可以大大简化管理员的 AT 分配负担。在我们的实验中，数据存储中的所有项都具有默认的安全级别。默认情况下，拒绝访问数据类型为 Person 的任何 URI 的个人信息都被保密。这可以防止代理对任何未被授予明确许可的个人做出推论。但是，如果授予代理对特定类型或属性的显式访问权限，就还应授予该代理对该类型或属性的子类型或子属性的默认访问权限。

例如，考虑一个谓词文件 Likes 列出个人喜欢的元素。假设 Jim 是一个人，他喜欢 Flying、SemanticWeb 和 Jenny，这几项分别是 Hobby、ResearchInterest 和 Person 类型的 URI 的个人信息，那么 1 是 ATT <1，Subject，URI，Jim>和<1，likes，Predicate，_>的 AT。默认情况下，只有 AT1 的代理商 Ben 不能从中得知 Jenny 在 Jim 的 Likes – list

中，因为 Jenny 的数据类型为 Person。但是，如果 Ben 还有 ATT < 2，Object，URI，Jenny > 描述的 AT2，那么 Ben 将能够在 Jim 的“喜欢”列表中看到 Jenny。

23.3.3　冲突

当发生以下 3 种情况时，我们认为将出现冲突：① 代理人拥有两个 AT，AT1 和 AT2；② AT2 的结果集是 AT1 的适当子集；③ AT1 的时间戳早于 AT2 的时间戳。在这种情况下，后者因更具体的 AT 取代了前者。所以 AT1 从 AT 列表中被丢弃以解决冲突。这种冲突出现在两个类型中，我们称之为子集冲突和子类型冲突。

当 AT2 是精简 AT1 的 ATT 的连接时，发生子集冲突。例如，假设 AT1 由 ATT < 1，Subject，URI，Sam > 定义，AT2 由 ATT < 2，Subject，URI，Sam > 和 < 2，Predicate，HasAccounts，_ > 定义。在这种情况下，AT2 的结果集是 AT1 的结果集的子集。因此，如果具有 AT1 的代理后来被分配到 AT2，则会发生冲突。当发生这种情况时，AT1 从代理的 AT 列表中被丢弃，来解决冲突。

当 AT2 中的 ATT 涉及数据类型是 AT1 中 ATT 的子类型时，发生子类型冲突。这里的数据类型，可以是主体、客体或两者结合的数据类型。

冲突解决可以总结成算法 1。这里，subset（AT1，AT2）是一个函数，如果 AT 的结果集合返回 true，那么 AT1 的结果集是 AT2 的结果集的真子集。如果 AT1 的主体是 AT2 的主体的子类型，则 SubjectSubType（AT1，AT2）返回 true。类似地，ObjectSubType（AT1，AT2）决定客体而不是主体的子类型关系。

算法 1：冲突检测与解决

输入：ATnewAT 与时间戳 TS_{newAT}

结果：检测冲突，如果不存在，将（newAT，TS_{newAT}）添加到代理的 AT 列表中

```
 1. currentAT[] ← the AT's and their timestamps;
 2. if (!Subset(newAT, tempATTS) AND
    !Subset(tempATTS, newAT) AND
    !SubjectSubType(newAT, tempATTS)) AND
    !SubjectSubType(tempATTS, newAT) AND
    !ObjectSubType(newAT, tempATTS)) AND
    !ObjectSubType (tempATTS, newAT)) then
 3.     currentAT[length_currentAT].AT ← newAT;
 4.     currentAT[length_currentAT].TS ← TS newAT;
 5. else
 6. count ← 0;
 7. while count < length_currentAT do
 8.     AT tempATTS ← currentAT[count].AT;
 9.     tempTS ← currentAT[count].TS;
10.     /* the timestamp during the AT assignment */
11.     if (Subset(newAT,tempATTS) AND (TS_newAT ≥ tempTS)) then
12.         /* a conflict occurs */
```

```
            currentAT[count].AT ← newAT ;
            currentAT[count].TS ← TSnewAT ;
        else if ((Subset(tempATTS,newAT))AND(tempTS < TS_newAT)) then
            currentAT[count].AT ← newAT ;
            currentAT[count].TS ← TSnewAT ;
        else if ((SubjectSubType(newAT, tempATTS) OR
                ObjectSubType (newAT,tempATTS)) AND TS_newAT≥tempTS) then
            /* a conflict occurs */
            currentAT[count].AT ← newAT ;
            currentAT[count].TS ← TSnewAT ;
    else if ((SubjectSubType(tempATTS, newAT) OR ObjectSubType (tempATTS,
newAT)) AND (tempATTS < TS_newAT)) then
            currentAT[count].AT ← newAT ;
            currentAT[count].TS ← TSnewAT ;
          end
          count ← count + 1;
       end
    end
```

23.4 系统架构

23.4.1 系统架构概述

我们提出的架构由两部分组成。图 23.4 的上半部分描述数据预处理组件，下半部分是负责回答查询的组件。

三个子组件将执行数据生成和预处理。我们使用 *N* – Triples 转换器组件将 RDF/XML（见本章参考文献［BECK04］）转换为 *N* – Triples 序列化格式（见本章参考文献［GRAN04］）。PS 组件获取 *N* – Triples 数据并将它们分解成谓词文件。本节介绍将这些步骤。最后一个组件输出后，用于收集汇总统计信息，这些统计信息被传递给 HDFS。

架构的底部显示了访问控制单元和 Map/Reduce 框架。访问控制单元参与查询执行的不同阶段。当用户提交查询时，重写该查询（如果可能的话）来强制执行一个或多个访问控制策略。Map/Reduce 框架有三个子组件。它需要从查询接口引擎获取重写的 SPARQL 查询，并将其传递给输入选择器和计划生成器。该组件选择输入文件，决定需要多少作业，并将信息传递给作业执行器组件，该组件将相应的作业提交给 Hadoop。作业执行器组件与访问控制单元通信以获得相关策略来执行，并相应地运行作业。然后它将查询答案从 Hadoop 中调出，送给用户。要回答需要推理的查询，我们要使用 Pellet OWL 推理器。这些策略存储在 HDFS 中，并在每次框架加载时由访问控制单元加载。

23.4.1.1 数据生成和存储

我们使用 LUBM（见本章参考文献［GUO05］）数据集进行实验。这个基准数据集

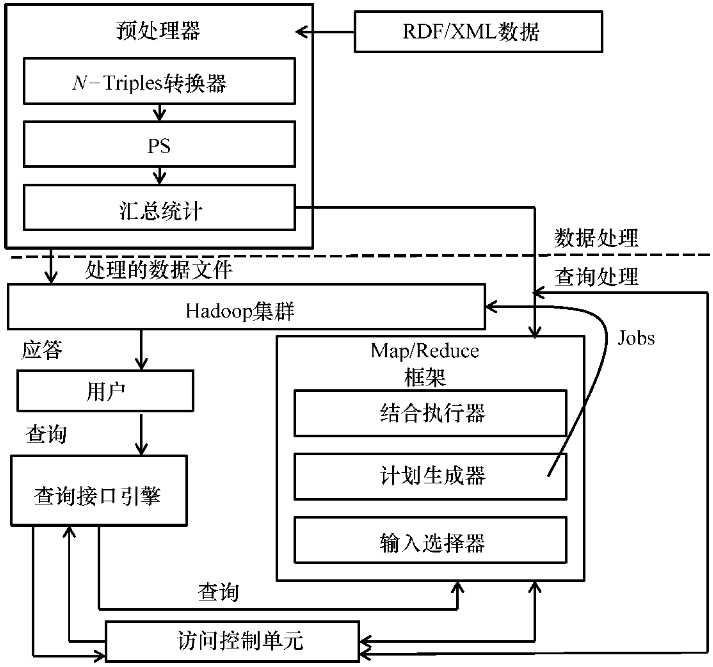


图 23.4　系统架构［源自 Khaled，A.，Husain，M. F.，Khan，L.，Hamlen，K. W. A Token－Based Access Control System for RDF Data in the Clouds，*Proceedings of the* 2010 *IEEE Second International Conference on Cloud Computing Technology and Science*（CloudCom），p. 104－111，2010. ©（2010）IEEE.］

被研究人员广泛使用（见本章参考文献［GUO04］）。LUBM 数据生成器以 RDF/XML 序列化格式生成数据。这种格式不适合我们的目标，因为我们将数据作为平面文件存储在 HDFS 中。如果数据采用 RDF/XML 格式，那么为了检索单个三元组，我们需要解析整个文件。此外，RDF/XML 格式不适合作为 Map/Reduce 作业的输入。因此，我们将数据存储为 *N*－Triples，因为使用该格式，我们在一个文件行中就具有完整的 RDF 三元组（主体、谓词和客体），这对 Map/Reduce 作业非常方便。但是，我们将数据转换为 *N*－Triples 格式，需通过谓词分割数据，这一步叫作 PS。在实际的 RDF 数据集中，不同谓词的数量不超过 10 或 20（见本章参考文献［STOC08］）。此分区减少了不包含变量谓词（见本章参考文献［PRUD08］）的任何 SPARQL 查询的搜索空间。对于这样的查询，我们可以为每个谓词选择一个文件，并仅在这些文件上运行查询。为了简单起见，我们用谓词命名文件，例如，包含谓词 p1：pred 的所有三元组都存储在名为 p1：pred 的文件中。关于这个过程的更详细的描述在本章参考文献［HUSA10］中给出。

23.4.1.2　示例数据

表 23.1 显示了三个谓词的示例数据。最左边的列显示 PS 步骤后学生客体的文件类型，它仅列出具有 rdf：type 谓词和学生客体的三元组的主体。其余的列在 PS 步骤之后

显示 advisor、takesCourse 和 teacherOf 谓词文件。每行都有一个主体－客体对。在所有情况下，可以从文件名中检索谓词。

表 23.1　LUBM 查询的样本数据

类型		ub：advisor		ub：takescourse		ub：teacherOf	
GS_1	Student	GS_2	A_2	GS_1	C_2	A_1	C_1
GS_2	Student	GS_1	A_1	GS_3	C_1	A_2	C_2
GS_3	Student	GS_3	A_3	GS_3	C_3	A_3	C_3
GS_4	Student	GS_4	A_4	GS_2	C_4	A_4	C_4
				GS_1	C_1	A_5	C_5
				GS_4	C_2		

来源：Khaled，A.，Husain，M. F.，Khan，L.，Hamlen，K. W. A Token－Based Access Control System for RDF Data in the Clouds，*Proceedings of the* 2010 *IEEE Second International Conference on Cloud Computing Technology and Science*（CloudCom），p. 104－111，2010. ©（2010）IEEE.

23.5　策略实施

我们的 Map/Reduce 框架分两阶段执行策略。在查询解析阶段只需重写 SPARQL 查询即可执行某些策略。剩余的策略可以通过两种方式在查询应答阶段实施。首先，当我们运行 Map/Reduce 作业以回答查询时，可以执行策略。其次，可以像查询一样执行作业，就像没有执行任何策略一样，然后输出并运行一组作业来执行策略。这些后处理的作业称为过滤作业。在这两种情况下，我们在输入选择器上选择输入文件时执行谓词级策略。在以下部分中，我们将详细讨论这些方法。

23.5.1　查询重写

可以通过重写 SPARQL 查询来执行涉及谓词的策略。这涉及用户有权访问的谓词替换成谓词变量。一个例子可以说明：假设用户的 AT 列表由 ATT < 1，predicate，takesCourses > 描述的 AT1 组成（即用户只能访问谓词文件 takesCourse）。如果用户在图 23.5 左侧提交了查询，我们可以用谓词文件替换谓词变量？p，重写的查询显示在图 23.5 的右侧。查询重写后，我们可以通过两种方式回答查询，详细介绍如下两部分。

SELECT ?o WHERE {A?*p* ?o}	⇒	SELECT ?o WHERE {A *takesCourse* ?o}

图 23.5　重写前后的 SPARQL 查询

23.5.2　嵌入式执行

在这种方法中，我们在 Hadoop 作业回答查询时执行策略，利用查询语言的连接机制来执行这种策略。可以通过这种方式实施涉及 URI、文字等的策略。例如，假设仅限于少数学生可以访问某些机密课程的数据。如果无特权用户希望列出学生所采用的课程，可以使用文件 takesCourse 加入列出机密课程的文件，从而在 Hadoop 作业的 Reduce

阶段执行所需的策略。假设课程 C_3 和 C_4 是保密课程。如果非特权用户希望列出 GS_3 所采用的课程，那么我们可以通过映射和规约算法 2 和 3 中所示的代码来回答查询。

算法 2：EEMAP 伪代码

```
1: splits ← value . split ()
2: if Input file = sensitiveCourses then
3:     output (splits [0], "S")
4:     else if splits [0] = GS3 then
5:     output (splits [1], "T")
6: end if
```

算法 3：EEREDUCE 伪代码

```
1: count ← 0
2: iter ← values . iterator ()
3: while iter . hasNext () do
4:   count + +
5:   t ← iter . next ()
6: end while
7: if count = 1 AND t = "T" then
8:   output (key)
9: end if
```

算法 2 显示了映射阶段的代码。它首先将每一行分成一个键和一个值。如果输入来自机密课程文件，则输出课程和一个标记（“S”表示“秘密”）。这个标记用来输出主体是否为第 4 行中的 GS_3。如果是，则输出课程就作为键和用于表示课程是学生 GS_3 的标记（“T”）。表 23.2 的左半部分显示了示例数据运行算法 2 的输出。

算法 3 显示了规约阶段的代码。它获得一个课程作为键和标志字符串作为值。表 23.2 的右半部分显示了示例数据运行算法 3 的输入。代码只是计算学生 GS_3 的数量（第 7 行），然后输出课程（第 8 行）。学生 GS_3 选取的保密课程有一个额外的标志，将计数提高到 2，防止这些课程被报告。学生未选取的机密课程也将有一个标志，表示这是一个机密课程。我们要检查学生 GS_3 所用课程的标志，是否防止这样的课程被报告。这两个检查一起确保 GS_3 所选取的非保密课程在结果中输出。因此，输出中只出现课程 C_1。

表 23.2　EEMap 输出和 EEReduce 输入

EEMap 输出		EEReduce 输入	
键	值	键	值
C_1	T	C_1	T
C_3	S	C_3	S，T
C_3	T	C_4	S
C_4	S		

来源：Khaled，A.，Husain，M. F.，Khan，L.，Hamlen，K. W. A Token－Based Access Control System for RDF Data in the Clouds，*Proceedings of the* 2010 *IEEE Second International Conference on Cloud Computing Technology and Science*（CloudCom），p. 104－111，2010. ©（2010）IEEE.

23.5.3 后处理执行

第二种方法如同没有访问控制一样运行作业，然后运行一个或多个其他作业，以根据策略过滤输出。这种方法的优点是实现起来很简单，但回答查询可能需要更长的时间。我们可以使用前面的例子来说明这种方法。我们首先运行这个作业，就好像对课程没有限制。然后，我们再执行一项作业来执行该策略。这个额外的作业需要两个文件作为输入：第一个作业的输出和包含机密课程的 URI 的 confidentialCourses 文件。在 Map 阶段，我们输出课程作为键，根据输入文件，输出一个标志字符串。映射代码与算法 2 大致相同。唯一的区别是，我们不需要检查标识学生的 URI，因为第一个作业的输出将包含只有该学生所选取的课程。Reduce 阶段的代码保持不变。因此，在第二阶段工作结束时，我们得到的输出结果中不包含任何机密课程。

23.6 实验配置和结果

我们在具有 10 个节点的 Hadoop 集群中运行了实验。每个节点具有 Pentium IV 2.80 GHz 处理器，4 GB 主内存和 640 GB 磁盘空间。操作系统是 Ubuntu Linux 9.04。我们将嵌入式执行方法与后处理执行方法进行了比较。使用 LUBM100、LUBM500、LUBM1000、LUBM2000、LUBM6000 和 LUBM9000 数据集进行实验。

我们使用两种场景来比较这两种方法：takeCourse 和 displayTeachers。在 takeCourse 方案中，无特权用户无法让任何学生查看机密课程列表。首先，提交一个查询，以显示一位特定学生所采取的课程。在 displayTeachers 场景中，非特权用户可以仅查看讲师的信息。通过提交查询，可以显示在特定部门受雇的人员的 URI。尽管教授、助理教授、副教授等在该部门受雇，但是由于策略，只有讲师的 URI 才能返回。详细结果见本章参考文献［KHAL10］。我们观察到，后处理执行总是比嵌入式执行方法多用 20% ~ 80% 的时间。这可以通过后处理所需的额外工作来轻松解释。无论作业的输入和输出数据大小如何，Hadoop 都需要大致相等的时间来设置作业。后处理执行方法比嵌入式执行方法需要运行更多的工作，从而产生了我们观察到的更多开销。

23.7 总结和展望

相关文献中已经广泛讨论了单台机器上的 RDF 数据访问控制，但是对于大型数据集，这些系统的规模处理能力仍较差。网络中的 RDF 数据量正在迅速增长，所以这是一个严重的制约。处理这些数据的最有效方法之一，是将其存储在云计算机中。然而，对于云驻留的 RDF 数据，访问控制尚未得到充分地应用。我们的实施机制结合了基于令牌的访问控制系统，系统用户可以根据业务需求和授权级别授予令牌。目前，我们正在建立一个包含令牌并能解决策略冲突的通用系统。目标是实现主体模型级访问，递归提取主体的客体，并将这些客体视为主体，只要这些客体是 URI。这将允许具有模型级

别访问的代理在给定主体上生成模型。

我们目前的工作还检查了混合云中的查询处理安全（见本章参考文献［OKTA12］）。未来，我们将探讨云查询处理的各种访问控制模型，并进行评估。这将使我们能更好地了解最适合云查询处理的访问控制模型。

参考文献

[APAC] Apache Hadoop. http://hadoop.apache.org/

[APAC] Apache Jena Fuseki. http://jena.apache.org/documentation/serving_data/

[BECK04] Beckett, D., RDF/XML syntax specification (revised). Technical report, W3C, February 2004.

[LEE98] Berners-Lee, T., Semantic web road map. http://www.w3.org/DesignIssues/Semantic.html, 1998.

[BOUG04] Bouganim, L., F.D. Ngoc, and P. Pucheral, Client-based access control management for XML documents. In: *Proceedings of the 20émes Journées Bases de Données Avancées (BDA)*, pp. 65–89, Montpellier, France, October 2004.

[BROE02] Broekstra, J., A. Kampman, and F. van Harmelen, Sesame: A generic architecture for storing and querying RDF. In: *Proceedings of the 1st International Semantic Web Conference (ISWC)*, pp. 54–68, Sardinia, Italy, June 2002.

[CHOI09] Choi, H., J. Son, Y. Cho, M.K. Sung, and Y.D. Chung, SPIDER: A system for scalable, parallel/distributed evaluation of large-scale RDF data. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pp. 2087–2088, Hong Kong, China, November 2009.

[GRAN04] Grant, J. and D. Beckett, RDF test cases. Technical report, W3C, February 2004.

[GUO04] Guo, Y., Z. Pan, and J. Heflin, An evaluation of knowledge base systems for large OWL datasets. In: *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pp. 274–288, Hiroshima, Japan, November 2004.

[GUO05] Guo, Y., Z. Pan, and J. Heflin, LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics*, 3(2–3): 158–182, 2005.

[HARR05] Harris, S. and N. Shadbolt, SPARQL query processing with conventional relational database systems. In: *Proceedings of the Web Information Systems Engineering (WISE) International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS)*, pp. 235–244, New York, NY, November 2005.

[HOLM99] Holmquist, L. E., J. Redström, and P. Ljungstrand, Token-based access to digital information. In: *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC)*, pp. 234–245, Karlsruhe, Germany, September 1999.

[HUSA09] Husain, M. F., P. Doshi, L. Khan, and B. M. Thuraisingham, Storage and retrieval of large RDF graph using Hadoop and MapReduce. In *Proceedings of the 1st International Conference on Cloud Computing (CloudCom)*, pp. 680–686, Beijing, China, December 2009.

[HUSA10] Husain M. F., L. Khan, M. Kantarcioglu, and B. Thuraisingham, Data intensive query processing for large RDF graphs using cloud computing tools. In: *Proceedings of the IEEE 3rd International Conference on Cloud Computing (CLOUD)*, pp. 1–10, Miami, Florida, July 2010.

[JAIN06] Jain, A. and C. Farkas, Secure resource description framework: An access control model. In *Proceedings of the 11th ACM Symposium on Access Control Models and Technologies (SACMAT)*, pp. 121–129, Lake Tahoe, California, June 2006.

[KHAL10] Khaled, A., M.F. Husain, L. Khan, K.W. Hamlen, and B.M. Thuraisingham, A Token-Based Access Control System for RDF Data in the Clouds. *IEEE Cloud Com*, Indianapolis, IN, December 2010.

[KIM08] Kim, J., K. Jung, and S. Park, An introduction to authorization conflict problem in RDF access control. In: *Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES)*, pp. 583–592, Zagreg, Croatia, September 2008.

[KOWA] Kowari. http://kowari.sourceforge.net.

[MIKA08] Mika, P. and G. Tummarello, Web semantics in the clouds. *IEEE Intelligent Systems*, 23(5): 82–87, 2008.

[OKTA12] Oktay, K.Y., V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra, and B. M. Thuraisingham, Risk-Aware Workload Distribution in Hybrid Clouds. *IEEE Cloud*, Honolulu, HI, June 2012.

[PRUD08] Prud'hommeaux, E. and A. Seaborne, SPARQL query language for RDF. Technical report, W3C, January 2008.

[REDD05] Reddivari, P., T. Finin, and A. Joshi, Policy based access control for an RDF store. In: *Proceedings of the Policy Management for the Web Workshop*, Chiba, Japan, 2005.

[STOC08] Stocker, M., A. Seaborne, A. Bernstein, C. Kiefer, and D. Reynolds, SPARQL basic graph pattern optimization using selectivity estimation. In: *Proceedings of the 17th International Conference on World Wide Web (WWW)*, pp. 595–604, Beijing, China, April 2008.

[W3C09] W3C. Semantic web frequently asked questions. http://www.w3.org/RDF/FAQ, 2009.

第 24 章　基于云的信息集成安全

24.1　概述

Amazon S3（见本章参考文献［AMAZ］）等云计算服务因成本效益和易维护等因素而受到广泛的欢迎。我们已经评估了使用 S3 存储服务，来使用情报社区的 Blackbook 系统存储语义 Web 数据的可行性。Blackbook 是 IARPA（情报高级研究计划局）针对构建基于语义 Web 的数据集成框架（见本章参考文献［BLAC］）的一项举措。Blackbook 系统的主要目的是为情报分析人员提供易于使用的工具，从不同的数据源访问数据，在数据源之间进行逻辑推理，并与使用该系统的其他分析人员分享这些知识。除了提供 Web 应用程序界面外，它还通过 Web 服务公开其服务。Blackbook 集成了来自不同数据源的数据，从而谨慎地将数据源存储在如云计算服务提供的共享环境中。Blackbook 基本上使用几种语义数据源来产生搜索结果。但是，以安全的方式在云环境中存储共享数据是一个巨大的挑战。本章将讨论解决这个问题的方法。

在我们的方法中，我们以安全的方式存储了 Amazon S3 上的一本 BlackBerry 数据源，从而在基于语义 Web 的框架中利用了云计算服务。我们使用高级加密标准（见本章参考文献［AES］）加密数据源，然后将其存储在 Amazon S3 上。此外，我们不会将原始密钥存储在系统的任何位置。相反，密钥由两个单独的组件生成，每个组件称为“密钥服务器”。然后，生成的密钥用于加密数据。

为了防止重放攻击，我们使用 Lamport 一次性密码（One Time Password，OTP）（见本章参考文献［LAMP81］）方案来生成客户端使用的密码，主要是为了“密钥服务器”认证。我们使用 RBAC 模型（见本章参考文献［SAND96］）来限制系统访问，授权用户使用 Sun 实现 XACML（见本章参考文献［OASI］）实施 RBAC 策略。

在本章中，我们将描述使用 Blackbook 的信息集成安全框架的设计和实现。本章的组织架构如下：在 24.2 节中，我们详细介绍框架的实现。24.3 节介绍实验结果。24.4 节中是关于本章总结与展望。我们工作的其他细节可以在本章参考文献［PARI09］和［PARI12］中找到。图 24.1 说明了本章包含的内容。Blackbook 的细节可以在本章参考文献［BLAC］中找到。

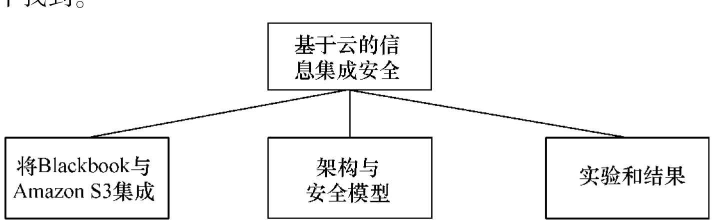


图 24.1　基于云的信息集成安全［Parikh，P.，M. Kantarcioglu，V. Khadilkar，B. M. Thuraisingham，and L. Khan：Secure information integration with a semantic web-based framework. *Proceedings of the IEEE 13th International Conference on Information Reuse and Integration*（*IRI*2012），Las Vegas，NV，659-663. ©（2012）IEEE.］

24.2 将 Blackbook 与 Amazon S3 集成

如前所述，基于语义 Web 的数据集成框架 Blackbook 允许从各种数据源中进行数据集成。我们发现 RDF 资源是通过 RESTful 的 Web 服务发布的最佳选择。由于 RESTful 的 Web 服务和语义 Web 都要处理资源，因此通过 RESTful 接口公开 RDF 资源是有意义的。语义 Web 等技术只能与使用 URI 识别资源的 Web 服务配合使用，因此，REST 是实现基于语义 Web 的系统的 Web 服务的理想平台（见本章参考文献［PARI09］）。

如第三部分所述，云计算是一种计算模式，其中扩展是动态的，通常是虚拟化的资源，是通过互联网提供的（见本章参考文献［CLOU］）。该概念包含以下组合：

■ IaaS（基础设施即服务）

■ PaaS（平台即服务）

■ SaaS（软件即服务）

经济优势是云计算模式背后的主要动机之一，因为它可以减少资本支出（Capital Expenditure，CapEx）和运营支出（Operational Expenditure，OpEx）（见本章参考文献［JENS09］）。各种组织可以使用云计算基础架构共享数据和计算能力。例如，salesforce. com 是 CRM 领域中的领导者，也是大规模利用云计算基础架构的先驱之一。由于 Blackbook 是一个数据集成框架，它可以搜索和集成来自可能位于本地计算机或远程服务器上的各种数据源的数据。我们利用 Amazon S3 提供的数据存储服务来存储 Blackbook 使用的数据源。

我们选择 Amazon S3 的原因如下：

■ 成本效益：每 GB 每月低至 0. 125 美元的存储价格。

■ 易于使用：可以通过 REST 和 SOAP Web 服务调用。

■ 可靠性：亚马逊是云计算的重要参与者，以提供可靠的云计算服务而闻名。

当前云计算系统的主要挑战之一就是隐私风险。也就是说，隐私是云计算服务在法律遵从和用户信任方面的重要关注。在本章参考文献［PEAR09］中，关于在设计，云计算服务时应如何考虑隐私问题作者提供了一些有趣的见解。本章参考文献［PEAR09］中说明的主要隐私风险包括：

■ 因为云服务用户被迫追踪或者根据自己的意愿提供个人信息。

■ 对于使用云服务的组织：不遵守企业策略，丧失声誉和信誉。

■ 对于云平台的实施者，存储在平台上的敏感信息的暴露，声誉和信誉的丧失。

■ 对于云平台上的应用程序提供商：法律不合规，声誉丧失。

■ 对于个人信息的数据主体暴露。

我们在实验中使用了 Amazon S3。如本章参考文献［AMAZ］所述，Amazon S3 是互联网的存储。它旨在使开发人员更轻松地进行全网规模的计算。Amazon S3 提供了一个简单的 Web 服务接口，可以随时从网络上的任何位置存储和检索任何数量的数据。它使任何开发人员均可以访问具有可扩展、可靠、快速、便宜的数据存储基础设施，这些

特征与亚马逊用来运行自己的全球网络的站点高度一致。该服务旨在最大化规模效益下，将这些优势传递给开发人员。

许多组织使用 Amazon S3 等服务进行数据存储。其中需要解决的一些重要问题包括：我们在 S3 上存储的数据是否安全？组织之外的任何用户都可以访问吗？我们如何限制组织内用户对文件的访问？为了保护我们的数据安全，我们建议在上传到 Amazon S3 数据文件之前，使用 AES 对数据进行加密。为了限制对组织内用户的文件访问，我们建议使用 XACML 实现基于角色的访问控制策略。在 RBAC 中，权限与角色相关联，用户成为相应角色的成员。这能简化权限的管理（见本章参考文献［SAND96］）。对应的系统架构如图 24.2 所示。

数据源以加密形式存储在 Amazon S3 服务器上。用于加密数据源的两个密钥存储在两个服务器：密钥服务器 1 和密钥服务器 2。与不同用户的数据源相关联的策略也存储在这些服务器上。

系统使用 OTP 进行认证，它是仅对单个会话或事务有效的密码。OTP 避免了与静态密码关联的缺点（见本章参考文献［ONE］）。与静态密码不同，它们不易受到重放攻击的影响。因此，如果入侵者设法掌握以前用于登录服务或携带事务的 OTP，则系统的安全性就不会受到损害，因为该密码将不再有效。OTP 的唯一缺点是人类无法记住它，因此需要额外的技术来实现。

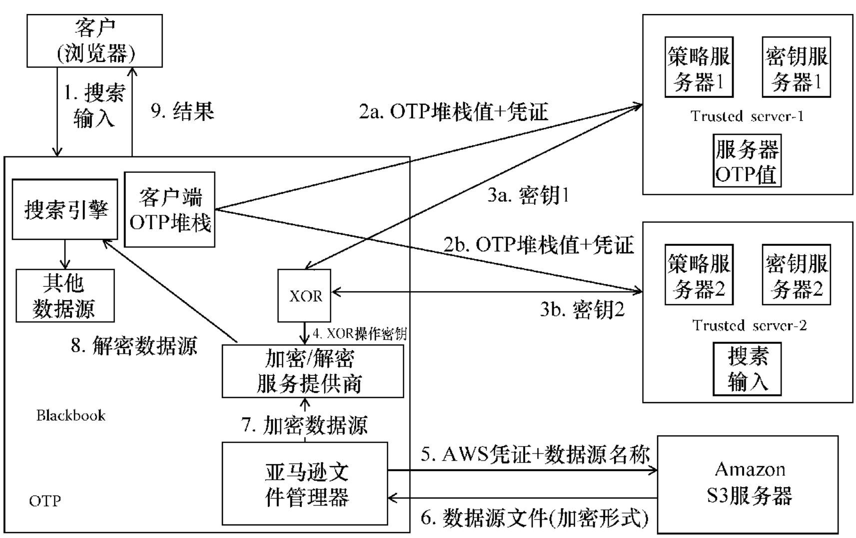


图 24.2　系统架构［Parikh，P.，M. Kantarcioglu，V. Khadilkar，B. M. Thuraisingham，and L. Khan. Secure information integration with a semantic web – based framework. *Proceedings of the IEEE 13th International Conference on Information Reuse and Integration*（*IRI*2012），Las Vegas，NV，659 – 663. ©（2012）IEEE.］

OTP 生成算法利用随机性，来防止基于以前观察到的 OTP 对未来 OTP 的预测。生成 OTP 的一些方法如下：

■ 根据以前的密码，使用数学算法生成新密码。

■ 基于认证服务器和提供密码的客户端之间的时间同步。

■ 使用基于挑战（例如，由认证服务器选择的随机数或事务细节）和/或计数器的新密码的数学算法。

我们使用 Lamport 的 OTP 方案进行身份验证。Lamport 的 OTP 方案是基于生成“密钥”值序列的数学算法，并且每个后续值基于前导的值。Lamport 的 OTP 方案的核心要求是，协作的客户端/服务器组件同意使用通用的排序算法，来生成一组即将到期的 OTP（客户端），并验证每个客户端发起的请求（服务方）中包含的客户端提供的密钥。在我们的例子中，客户端是 Blackbook 系统，服务器组件是“密钥服务器”。客户端生成从“种子”值开始的有限序列的值，并且每个后继值通过应用一些变换算法［或 F（S）函数］得到先前的序列值：

S1 = seed，S2 = F（S1），S3 = F（S2），S4 = F（S3）…S［n］ = F（S［$n-1$］）

我们使用一些随机生成的字节（使用 SHA1PRNG）对用户加盐后的“密码”作为键，使用 SHA－256（见本章参考文献［SECU02］）生成“种子”值。使用 SHA－256 所获得的“种子”值生成序列中的下一个值。所有这些生成的值都将存储在客户机上的堆栈中。堆栈中最上面的值存储在“密钥服务器”（1&2）上。如果客户端首次发送请求，则将客户端堆栈的最高位置的值与“密钥服务器”（1&2）上的值进行比较。如果值匹配，客户端将进行身份验证，并删除客户端堆栈上的最高位置的值。对于后续请求，客户端堆栈上的最高位置的值用于使用哈希函数（用于构建堆栈）计算后续值。如果生成的值和“密钥服务器”上的值匹配，则认证用户，客户端堆栈上的最高位置的值存储在“密钥服务器”上，随后从客户端堆栈中删除。如果客户端堆栈耗尽，将生成一个新的堆栈，并且堆栈上的最高位置的值仍存储在“密钥服务器”上。一旦用户使用 OTP 方案进行身份验证，就根据适用于资源（本例中的数据源）的策略对用户请求进行评估并由用户请求访问。预定义的策略存储在“密钥服务器”的“策略服务器”组件中。如果资源的策略适用于用户请求，“密钥服务器”将发送用于加密用户请求资源的密钥。

我们使用 XACML 来实现使用 XML 文件中定义策略的访问控制。在用户通过系统认证后，系统检查用户是否被授权访问所请求的资源。用户请求由 PEP 处理，将用户的请求转换为 XACML 请求，并将其发送到 PDP 进行进一步评估。PDP 对请求进行评估，并发送响应，该响应可以是“允许访问”或“访问被拒绝”，并附带相应的职责。（我们没有考虑到我们系统的职责。）一个策略可以是以下几个子组件的集合：目标、规则、规则组合算法和职责。

目标：每个策略只有一个目标，有助于确定策略是否与请求相关。策略与请求的相关性决定了是否对请求进行评估，这是通过定义目标主体、资源和操作中三个类别的属性来实现的。例如，我们为主体指定了值“testadmin@ blackbook. jhuapl. edu”，为资源指定了“Amazons3”。

规则：通过规则，我们可以将多个角色与策略相关联。每个规则由条件、效果和目

标组成。

条件是关于在评估时返回 true、false 或不确定的属性的语句。

效果是对要满足的规则假设允许或者拒绝而产生的后果。我们已经将该值指定为允许。

目标有助于确定某一规则是否与请求相关。

规则组合算法：由于策略可以有各种规则，不同的规则可能产生冲突的结果。规则组合算法要解决这种冲突，以便每个请求、每个策略获得一个结果。只有一种规则组合算法适用于一种策略。

职责：职责允许相应机制提供更精细的访问控制级别，而不仅仅是允许和拒绝决策。它们是 PEP 必须执行的行动，同时执行授权决定。

成功的认证和授权后，Amazon 文件管理器会从 Amazon S3 服务器下载所请求的资源。更具体地说，密钥服务器 1 发送 key1，密钥服务器 2 发送 key2 给亚马逊文件管理器。通过对这些密钥进行 XOR 操作来获取 keyorg，也就是说，

keyorg = key1 XOR key2

然后，加密/解密服务提供商使用 keyorg 来解密资源。

使用两个密钥服务器的主要动机是避免单一故障。如果任何密钥服务器被黑客入侵，则数据不会因为两个密钥而受到影响，每个密钥服务器都需要一个密钥来解密数据源。如果其中一个密钥服务器遭到入侵，并且存储在该服务器上的密钥受到威胁，则我们就会遇到存储在 Amazon 上的数据源将无法使用的风险，因为我们需要两个密钥（每个密钥服务器中的一个密钥）来检索原始密钥，用于加密数据源。为了避免这种情况，我们建议定期备份每个密钥服务器上的密钥。

场景：我们现在将描述一个示例场景，其中描述了与 Amazon S3 存储服务的交互，也涉及 Blackbook 系统：

1）用户 U 向 Blackbook 发送搜索查询（图 24.2 中的步骤 1）。Blackbook 联合各种数据源的查询，包括安全存储在 Amazon S3 上的数据源 F。

2）遵循 OTP 方案来验证客户端（在这种情况下为 Blackbook），以使用 AWS S3 服务。客户端机器将 OTP 堆栈上的最高位置的值与用户凭证和请求，一起发送到密钥服务器 1&2（图 24.2 中的步骤 2a 和 2b）。

3）如果客户端传递的值与密钥服务器的 OTP 堆栈值相匹配，并且适用于用户的策略，针对请求也是有效的，则密钥服务器发送用于解密数据源的“密钥”（在图 24.1 中步骤 3a 和 3b ）。

4）从密钥服务器 1&2 中获得密钥 key1 和 key2，然后进行 XOR 操作，以此来获得用于解密数据源 F 的原始密钥（图 24.2 中的步骤 4）。

5）亚马逊文件管理器传递 Amazon 账户凭证和数据源名称，以检索数据源（图 24.2 中的步骤 5 和 6）。

6）加密/解密服务管理器检索加密的数据源，然后使用 XOR 后的密钥解密数据源（图 24.2 中的步骤 7 和 8）。

7）Blackbook 对从 Amazon 获取的数据源和其他数据源进行搜索，并将结果返回给用户（图 24.2 中的步骤 9）。

示例 XACML 请求：属于用户组（主体的属性）的一个主体 testadmin@blackbook.jhuapl.edu 尝试对资源 Amazons3 执行读取操作。要创建这样一个请求，我们需要两个主体属性：一个资源属性和一个动作属性。两个主体属性是 rfc822Name（电子邮件 ID）和主体所属的组。一个资源属性是资源的 URI，一个动作属性是资源上的读取操作。展示 PEP 和所有这些属性生成的完整列表，在本章参考文献［PARI09］中找到。

24.3 实验

在我们的方法中，我们已经使用 AES 将数据存储在 Amazon S3 服务器之前进行加密。在 Amazon 服务器中上传数据是一次性过程。仅当存储的数据需要修改时，才需要重新上传数据源。但是，每当用户向 Blackbook 系统发出搜索查询时，均需要下载存储在 Amazon S3 上的数据源。由于数据源需要在每次发出查询时进行解密，所以可能会影响性能，因为加密和解密是开销昂贵的操作。

我们在运行 Ubuntu Gutsy 7.10 的戴尔台式机上进行了实验，硬件配置如下：Intel® Pentium®4 CPU 3.00GHz，1 GB RAM。运行实验时的网络带宽在 250 和 300Mbit/s 之间变化。我们使用由 SP2B SPARQL Performance Benchmark（见本章参考文献［SPAR］）提供的三元组程序生成数据文件。我们尝试了 30 个不同大小的文件，范围从 1 ~ 30 MB。实验细节在本章参考文献［PARI09］中给出。

24.4 总结和展望

云计算模式在当今世界越来越重要。因此，云计算背景下的数据安全和隐私等问题已经引起了人们的广泛关注。在本章中，我们描述了在存储到如 Amazon S3 之类的云计算服务器之前，对其进行加密来保护我们数据的技术。我们的方法是新颖的，因为我们建议使用两个密钥服务器来生成和存储密钥。此外，我们确保比一些其他已知方法更安全，因为我们不存储用于加密数据的实际密钥。即使一个或两个密钥服务器受到威胁，这也可确保对我们数据的保护。我们的实验使用 Blackbook，这是一种基于语义 Web 的数据集成框架，它集成了各种数据源的数据。

在目前的方法中，我们从用户的每个请求中下载数据源。在将来，我们可以提供缓存用户在本地服务器上请求的数据源，并通过使其视为本地数据源将缓存数据源的结果提供给用户。这种方法将有助于提高性能。此外，我们也可以将数据源划分成块，然后上传。由于搜索过程以异步方式进行，因此我们可以下载这些块，再搜索结果，并将它们一次显示给用户。同时，应用程序可以继续下载其他块。我们还可以跟踪用户的搜索记录。当用户登录系统时，我们可以下载用户最有可能查询的块。

参考文献

[AES] Advanced encryption standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[AMAZ] Amazon S3, http://aws.amazon.com/s3/

[BLAC] http://info.publicintelligence.net/IARPA_overview_UMD.pdf

[CLOU] Cloud computing, http://en.wikipedia.org/wiki/Cloud_computing

[LAMP81] Lamport, L., Password authentication with insecure communication. *Communications of the ACM*, 24(11): 770–772, 1981.

[JENS09] Jensen, M., J. S. Jensen, N. Gruschka, and L. L. Iacono, On technical security issues in cloud computing, *Proceedings of IEEE International Conference on Cloud Computing*, Bangalore, India, 2009.

[OASI] OASIS, http://www.oasis-pen.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf

[ONE] One time password, http://en.wikipedia.org/wiki/One-time_password

[PARI09] Parikh, P., Masters thesis, *Secured Information Integration with a Semantic Web-Based Framework*, 2009, the University of Texas at Dallas.

[PARI12] Parikh, P., M. Kantarcioglu, V. Khadilkar, B. M. Thuraisingham, and L. Khan, Secure information integration with a semantic web-based framework. *Proceedings of the IEEE 13th International Conference on Information Reuse and Integration (IRI2012)*, Las Vegas, NV, 659–663.

[PEAR09] Pearson, S., HP labs—Taking account of privacy when designing cloud computing services, *Proceedings of IEEE ICSE Cloud09*, Vancouver, IEEE, May 2009.

[SAND96] Sandhu, R., E. J. Coyne, H. L. Feinstein, and C. Youman, Role based access control models. *IEEE Computer*, 29(2): 38–47, 1996.

[SECU02] Secure hash standard, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf, 2002.

[SPAR] SPARQL performance benchmark http://www.openlinksw.com/dataspace/vdb/weblog/vdb%27s%20BLOG%20%5B136%5D/1423.

第六部分总结

本部分的章节介绍了我们开发的云安全实验系统，包括云查询处理安全和信息集成安全。

在第 22 章中，我们提出一个允许合作组织安全地共享大量数据的系统。我们通过使用 Hadoop 来确保组织有一个很大的通用存储区域。此外，我们使用 Hive 向系统的用户呈现数据的结构化视图，并且还使得它们能够以类似 SQL 的语言来查询数据。

在第 23 章中，描述了存储在云中的语义 Web 数据的访问控制机制。我们实施的机制结合了基于令牌的访问控制系统，系统的用户可以根据业务需求和授权级别授予令牌。

在第 24 章中，我们描述了在存储到 Amazon S3 之类的云计算服务器上时，通过对其进行加密来保护我们数据的技术。我们的方法是新颖的，因为我们建议使用两个密钥服务器来生成和存储密钥。此外，我们提供了比一些其他已知方法具有更多的安全性，因为我们不存储用于加密数据的实际密钥。即使一个或两个密钥服务器受到威胁，这也可确保对数据的保护。

第七部分　面向安全应用的云实验系统

第七部分简介

在第六部分中，我们讨论了云计算安全系统的原型，在第七部分中，我们将讨论云计算在安全功能方面的应用。也就是说，我们将描述我们开发的系统，说明云系统如何提供安全即服务。

第七部分由 4 章组成：第 25 章、第 26 章、第 27 章、第 28 章。基于云的恶意软件检测系统将在第 25 章中讨论。基于云的内部威胁检测将在第 26 章中讨论。我们研究的云中信息共享系统在第 27 章中讨论。我们在第 28 章中描述基于语义云的信息共享系统的设计和实现。语义云是向消费者提供语义 Web 服务的云。

第 25 章　面向演进数据流的基于云的恶意软件检测

25.1　概述

恶意软件是每年众多网络被攻击成功的重要工具，包括数据以及身份的窃取，系统以及数据的损坏和拒绝服务。因此这对许多个人和组织来说是一个巨大的安全威胁。从 1999 年到 2006 年，全球范围内由恶意软件平均每年造成的直接损失高达 140 亿美元（见本章参考文献［COMP07］）。这些损失包括对系统分析修复以及杀毒所产生的劳动力成本、生产力损失、系统损失或性能下降导致的收入损失，以及由于攻击而直接产生的其他成本。但是，直接经济损失不包括预防成本，比如杀毒软件、硬件以及 IT（信息技术）安全人员的工资等。除了这些货币损失之外，一旦被成功攻击，个人以及组织可能经常遭受身份盗取、信息盗取和其他无形的损失。

恶意软件包括病毒、蠕虫、特洛伊木马、时间和逻辑炸弹、僵尸网络以及间谍软件。研究人员已经制定了一些技术来应对这些攻击。但是，当研究人员针对攻击而预防和检测成功的次数越来越多时，更加复杂的恶意软件代码也随之出现了。因此，恶意软件编写者与恶意软件防御者之间的竞赛将持续升级。签名检测是一种被杀毒界广泛应用的恶意代码检测技术。这种技术需要将不可信任的可执行文件，与唯一的标识字符串或称为签名的字节模式相匹配，这个字符串被用作特定恶意代码的标识符。虽然签名检测技术被广泛使用，但是这项技术对 0 - day 漏洞攻击（新型恶意代码）、多态性攻击（同样二进制的不同加密）、变质攻击（同样功能的不同代码）并不是有效的（见本章参考文献［CRAN05］）。因此为了预防这些攻击，我们现需要一种更快速、自动化、高效且健壮的检测技术。

本章介绍一种专用于自动生成签名来防御这些攻击的数据挖掘技术。由于每时每刻都需要运行恶意软件检测工具，因此我们开发了基于云的数据挖掘工具。我们将在本章的其余部分中描述这种基于云的工具的详细设计和实现。

本章结构如下：25.2 节介绍恶意软件检测。25.3 节介绍相关工作。25.4 节讨论分类算法以及分析证明其有效性。25.5 节介绍为了恶意软件检测，而进行的基于云计算的特征提取和选择技术。25.6 节讨论数据采集、实验配置、评估技术和结果。25.7 节讨论与我们的方法有关的几个问题。25.8 节总结我们的方法。图 25.1 说明了本章包含的概念。

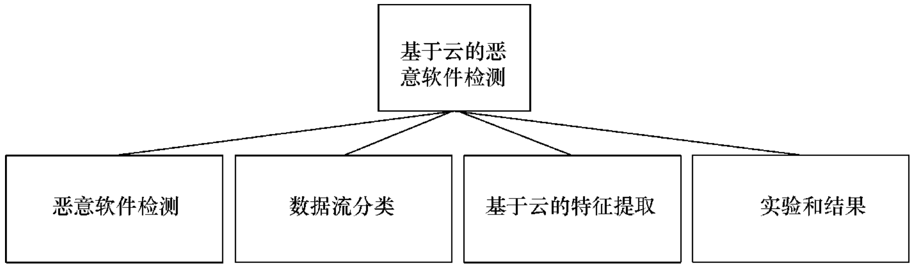


图 25.1　面向演进数据流的基于云的恶意软件检测

25.2　恶意软件检测

25.2.1　作为数据流分类问题的恶意软件检测

使用数据挖掘检测恶意软件的问题（见本章参考文献［SCHU01］、［KOLT04］、［MASU08a］）涉及将每个可执行文件分为良性或恶意程序。过去的大多数工作把这个问题处理为静态数据分类问题，其中分类模型是用固定训练数据训练的。但是，恶意软件的不断进化、创新和升级的速度并不适合静态训练。检测不断更新的恶意软件应该被更好地处理为数据流分类问题。在这种模式中，数据流是一连串可执行文件且每个数据点就是一个可执行文件。数据流是无限长的，而且这种方式还能观察到概念漂移。因为攻击者不断地开发新的技术以及改变恶意代码的特征，以避免被检测。同样，良性可执行文件的特性也随着编译器与操作系统的演进而改变。

数据流分类是数据挖掘网络交流社区积极研究的主要领域，需要克服至少 3 个挑战：第一，为了训练目的，在无限长的概念漂移中，存储和维护潜在的无限历史数据是不可行的。第二，分类器模型必须不断适应数据流的概念漂移。第三，如果数据流中没有数据点相对应的预定义特征空间，那么随着数据流的演进，新的具有高识别力的特征就必须被选择和提取，这被称为特征演进。

前两个问题的解决方案是相关联的。概念漂移需要改进假设以适应新概念，大多数旧的数据必须从训练数据集中丢弃。因此，挖掘概念漂移数据流中一个主要问题是选择适当的训练实例来学习概念的演进。解决第三个问题需要一个持续的特征选择过程，因为新的和更强大的特征可能会出现，旧的特征也可能随着概念的演变而变得不那么有优势。如果特征空间足够大，则特征提取和选择的运行时间和内存要求会成为数据流分类系统的瓶颈。

解决概念漂移的一种方法是选择和存储那些最符合目前概念的训练数据集（见本章参考文献［FAN04］）。另外一些方法，比如超快速决策树（Very Fast Decision Trees，VFDT）（见本章参考文献［Domingos 和 HULT00］），当新数据出现时更新现有的分类模型。但是过去的研究表明，集成技术在处理突发变化和概念漂移时展现出强大的稳定性（见本章参考文献［WANG03］、［SCHO05］、［KOLT05］）。这些技术都能维持一个分

类器的集成，并在新的数据出现时更新该集成。

我们设计并开发了一种多分区、多块集成的分类算法，这种算法泛化了现有的集成方法。通过对真实数据流的测试证明，这种泛化相对于现有的单分区、单块集成方法来说显著提高了分类的准确性。在我们方法中的集成包括 Kv 个分类器，K 是一个常数，v 是分区的数量。我们稍后将会解释它们。

我们的方法将数据流分成相等大小的块。块的大小通过每块数据都应适合主内存来确定。被标记的每个块都被用来训练分类器。当新的数据块被标记时，集成也跟着被更新。我们将 r 个最近标记的连续数据块分为 v 个分区，并且每个分区都被用来训练分类器。因此，r 个连续数据块用于训练 v 个分类器，然后，通过在新训练的 v 个分类器和现有的 Kv 个分类器之间选择最好的 Kv 个分类器（基于准确性），来更新集成。所以集成中分类器的总数将保持不变。因此我们的方法通过分区数 v、块数 r 以及集成大小 K 来参数化。

我们的方法不再假设流中出现的新数据点立即被标记。相反，它会延迟整体更新过程，直到最新数据块中数据点的标签变得可用。与此同时，新的未被标记的数据被现有的集成持续分类。因此，该方法非常适合于错误分类从专家用户或其他来源请求校正标签的应用。例如，考虑到在线信用卡欺骗检测问题。当一个新的信用卡发生交易时，它的分类（欺诈或真实）是使用当前的集成预测的。假设一个欺诈交易被误判为真实交易，当客户收到银行对账单时，他会识别该错误并向当局报告。通过这种方法，获得数据点的实际标签并且相应更新集成。

25.2.2 将云计算用于检测恶意软件

如果数据点的特征空间不是固定的，则分类问题的子问题是描述每个数据点特征的选择和提取。在早期的工作（例如，本章参考文献［KOLT04］），我们使用二进制 n – grams 作为恶意软件检测特征。但是，由于 n – grams 总数可能非常大，我们会明智地选择那些具有最好鉴别力的 n – grams。这个选择的过程是持续的。随着数据流的演进，新的 n – grams 将出现并且要比旧的 n – grams 更占据主导地位。在我们的模型中，为了在特定的时期确定最好的特征，这些新的 n – grams 往往要取代旧的 n – grams。

对于大型数据集来说，单纯实现特征的提取与选择过程，既可以基于时间密集型也可以基于存储密集型。例如，我们在之前的工作中（见本章参考文献［MASU08a］），从语料库中提取的 25000 万个 n 连字串中只有 3500 个可执行文件，这种特征提取的过程需要大量的虚拟内存开销（相关性能开销），因为不是所有的特征都能存储在主存储器。提取和选择特征对一个四核处理器和 12GB 内存的计算机来说，需要大约两个小时的计算时间和多个 GB 的磁盘空间。况且这是使用纯静态数据集，但当数据集是动态流时，提取和选择必须重复，这将会面临一个重大难题。本章中，我们考虑包含 10.5 万个可执行文件的更大数据集，这对我们之前的方法来说是不可能胜任的。

因此，我们设计和开发了可扩展的选择和提取方案，主要是利用了云计算框架（见本章参考文献［DEAN08］）。我们也说明了，根据集群节点的可用性，特征提取和

选择的运行时间可以在一定程度上减少，减少的原因在于因子 m，m 是云集群中的节点数。节点是廉价硬件商品的机器。因此，解决方案也是性价比较高的，因为不需要使用高端计算设备。

25.2.3　我们的贡献

我们的贡献可以概括如下。我们设计并开发了一种广泛的多分区、多块集成技术，明显降低了现有单分区、单块集成方法的预期分类误差。并通过理论分析证明了该方法的有效性。然后，我们将恶意软件检测问题定为数据流分类问题，并证明出传统的恶意代码检测技术相对于我们的数据挖掘方法的缺点。

我们针对这个问题利用云计算框架设计开发了一种可扩展和具有高性价比的解决方案。最后，将我们的技术应用于综合生成的数据，以及真正的僵尸网络传输和真实的恶意可执行软件，结果获得了比其他流数据分类技术更好的检测精度。结果表明，在面向基于数据流分类的入侵检测中，我们的集成技术是功能强大的工具。

25.3　相关工作

我们的工作与恶意软件检测和数据流挖掘技术相关。这两部分都会在本章讨论。传统的基于签名的恶意软件检测器，通过扫描不可信的二进制文件，来区分字节序列或特征来检测恶意软件。在特征数据库中维护恶意软件特有特征，必须在新的恶意软件被发现和分析的情况下不断更新。传统意义上，当新的恶意软件出现并被分析后，人类专家手动派生、更新和传播签名数据库。但是，在过去十年来，新的恶意软件出现率逐步提升和自我突变，多态恶意软件的出现使手动签名更新变得不切实际。从而推动了自动化数据挖掘技术来检测恶意软件（例如，本章参考文献［KOLT04］、［SCHU01］、［MASU08a］、［HAML09］）。这种方法能够自动推断出以往看不到的恶意软件的签名。

基于数据挖掘方法来分析可执行文件的内容，并且当发现特定的特征组合时将其归类为恶意软件。首先通过训练这些恶意软件检测器，来让它们可以归纳出恶意或良性可执行文件之间的区别，从而检测出未来恶意软件的实例。训练过程包括特征提取和利用特征建模。基于数据挖掘技术的恶意软件检测器的不同之处，在于区分特征是如何提取和利用哪种机器学习技术建立模型。这些技术的性能在很大程度上取决于被提取特征的质量。

在本章参考文献［SCHU01］的论述中，作者从 Windows PE 可执行文件的标题，以及可执行内容的 2 字节序列中，提取 DLL 调用信息（使用 GNU binutils）和字符串（使用 GNU 字符串）。这些 DLL 调用、字符串和字节被用来作为特征进行训练模型。使用两种不同的机器学习技术来训练模型：RIPPER（见本章参考文献［COHE96］）和朴素贝叶斯（Naïve Bayes，NB）（见本章参考文献［MICH94］），目的是为了比较它们的相对性能。见本章参考文献［KOLT04］中，作者从可执行文件中提取二进制 n - gram 的特征，并将其应用到不同的分类方法中，比如 K - 邻接（K - Nearset Neighbor，KNN）（见本章参考文献［AHA91］、NB、支持向量机（Support Vector Machines，SVM）（见本章

参考文献［BOSE92］）、决策树（见本章参考文献［QUIN03］）和 Boosting 算法（见本章参考文献［FREU96］）。Boosting 算法与其他各种学习算法相结合以获得改进的模型（比如增强决策树）。我们之前基于数据挖掘检测恶意软件的工作（见本章参考文献［MASU08a］），主要是从可执行文件中提取二进制 n – grams，从反汇编的可执行文件中提取汇编指令序列，并从程序头中提取 DLL 调用信息。在此项工作中使用的分类器模型是 SVM、决策树、NB、增强决策树和增强 NB。

Hamsa 和 Polygraph（见本章参考文献［LI06］、［NEWS05］）应用一种简单的数据挖掘形式，自动生成使用二进制 n 连字串作为特征的蠕虫签名。如果标签仅仅出现在恶意二进制文件（即阳性）中而没有出现在良性文件中，就确定一组 n – grams 为蠕虫标签。这在两个重要方面不同于我们之前讨论的传统的数据挖掘技术（包括我们的）：第一，Hamsa 和 Polygraph 将注意力局限在仅仅出现在恶意软件池中的 n – grams，相反，传统数据挖掘技术也考虑出现在良性池中的 n 元模型来提高分类准确性。第二，Hamsa 和 Polygraph 通过简单存在一组 n – grams 来定义签名匹配，而传统的数据挖掘方法构建了基于特征存在和不存在的样本匹配的分类模型。传统数据挖掘技术因此推广了 Hamsa 和 Polygraph 的方法，并且也相应增加了功能。

过去所有的工作几乎都已经将恶意软件检测当作静态数据分类问题进行了处理，并且使用固定的训练数据分类器模型。然而，随着恶意软件作者采用的新型恶意软件和新的混淆策略的迅速出现，引进了一个违反静态范例问题的动态组件。因此我们认为，有效的恶意软件检测必须越来越多地被当作数据流分类问题来看待，以跟上攻击技术发展的步伐。

许多现有的数据流分类技术针对的是无限长度的数据流，这些数据流显示有概念漂移（见本章参考文献［AGGR06］、［WANG03］、［YANG05］、［KOLT05］、［HULT01］、［FAN04］、［GAO07］、［HASH09］、［ZANG09］）。所有的这些技术采用单向增量更新方法，但是却使用不同的增量更新机制。大多数可以分为两大类：单模型增量方法和混合批量增量方法。

单模型增量更新动态更新单个模型时，涉及每个新的训练实例。例如，决策树模型可以根据输入数据增量更新（见本章参考文献［HULT01］）。相反，混合批量增量方法使用传统批量处理学习技术，从一批训练数据构建每个模型。旧的模型随着较新模型的出现定期被替换（见本章参考文献［WANG03］、［BIFF09］、［YANG05］、［FAN04］、［GAO07］）。一些混合方法使用一种单一模型分类未标记数据（例如本章参考文献［YANG05］和［CHEN08］），而其他则使用集成模型（例如本章参考文献［WANG03］和［SCHO05］）。混合方法的优点是模型更新通常比单模型方法简单得多。例如，集成时的分类器可以简单地被移除或更换。然而，通过逐步更新集成时的分类器，来组合这两种方法的其他技术可能相对更复杂（见本章参考文献［KOLT05］）。

精度加权分类器集合（Accuracy Weighted Classifier Ensembles，AWE）（见本章参考文献［WANG03］、［SCHO05］）是混合增量式集成分类器的一个重要类别，它通过加权多数投票进行分类。通过将数据流切割成相同大小的块，每个块用于训练一个分类器模型。包含 K 个这样模型的集成来分类未标记的数据。每当一个新的数据块被标记，

将从该块中训练一个新的分类器。这个分类器取代集成中一个现有的分类器。通过评估每个分类器在最新训练块上的准确性来选择被取代的分类器。这些集成方法的优点是可以比不断更新的单一模型更有效地构建它们，并且它们也可以比单一模型具有更高的精度（见本章参考文献［TUME96］）。

我们的集成方法与 AWE 分类器密切相关，但仍有一些显著差异。首先，我们采用训练数据的多分区来构建来自训练数据的 v 个分类器。第二，训练数据由 r 个连续数据块（即分块方法）而不是一个单一数据块组成。我们已经在分析和经验上证明了这两个功能的加强，即多分区和多块，明显降低了整体分类误差（见本章参考文献［MASU11］）。第三，当我们更新集合时，v 个新训练的分类器代替 v 个旧的分类器。被替换的 v 个分类器可能来自不同的块。因此，虽然某个块中的分类器从集成中被移除，但是块中的其他分类器仍可能存在于集成中。这不同于 AWE，其中删除分类器意味着完全删除从一个整块中获得的知识。我们的替换策略也有助于减少误差。最后我们采用简单的多数投票，而不使用加权投票，因此这样更适用于数据流，如本章参考文献［GAO07］所论述的那样。因此，我们的多分区、多块集成方法是由 AWE 实施的更为普遍和有效的形式。

我们之前就已经公开了在这个工作上所进行的扩展（见本章参考文献［MASU09］）。大多数现有的数据流分类技术，包括我们以前的工作都会假设数据流中数据点的特征空间是固定的。但是，在一些情况下，如文本数据，这一假设是无效的。例如，当特征是单词时，由于新单词频繁出现，因此在流的开头处不能完全确定特征空间。另外，很可能词典中的大部分词具有较低的辨别力，所以从特征空间中最好去掉它们。这样做，对每个数据点选择候选特征的子集来说，会更高效快捷。随着更具有鉴别能力的候选特征的出现和旧特征的过时，这种特征选择必须进行增量更新。因此特征选择和提取应该是数据流分类的一个重要组成部分。本章中，我们使用云计算框架设计和实施一个高效可扩展性的特征提取和选择技术（见本章参考文献［ZHAO09］、［DEAN08］）。这种方法取代了我们以前的工作，因为它考虑了当特征空间无法预料时，数据流分类中所产生的真正挑战。这有利于应用我们的技术，在不断变化的大数据集中检测真实可执行的恶意软件，从而显示恶意软件实例。随着时间的演进，各种较新的恶意软件也可以被检测出来。

25.4　系统设计与实现

25.4.1　集成结构与更新

我们的扩展、多分区、多块（Extended，Multi－Partition，Multi－Chunk，EMPC）集合学习方法维护最新的、最好的 Kv 个分类器的一个集合 $A = \{A_1, A_2, \cdots, A_{Kv}\}$。每当一个新的数据块 D_n 到达，将使用集合 A 测试该数据块。当块 D_n 被标记时，集合就被更新。分类过程采用简单的多数表决方式。

集成结构更新过程如图 25.2 所示，并总结成算法 1。算法的第 1～3 行计算块 D_n 上每个分类器 $A_i \in A$ 的错误，其中 D_n 是已标记的最近的数据块。令 D 是最近标记的 r 个数据

块的数据，包括 D_n。第 5 行随机把 D 平均分为 v 个部分 $\{d_1, \cdots, d_v\}$，所有的部分都有大致相同的类分布。第 6 ~9 行训练一组新的 v 个分类器，每个分类器A^n被数据集 $D-d_j$ 训练，使用每个分类器对应的测试数据来测试每个分类器$A^n \in A^n$的错误率。最终，第 10 行基于每个分类器第 2 ~8 行计算的错误率，从 $A^n \cup A$ 中的（$Kv+v$）个分类器中来挑选最佳 Kv 个分类器。注意到第 n 组 v 个分类器中任何子集都可能被挑选进入新集合。

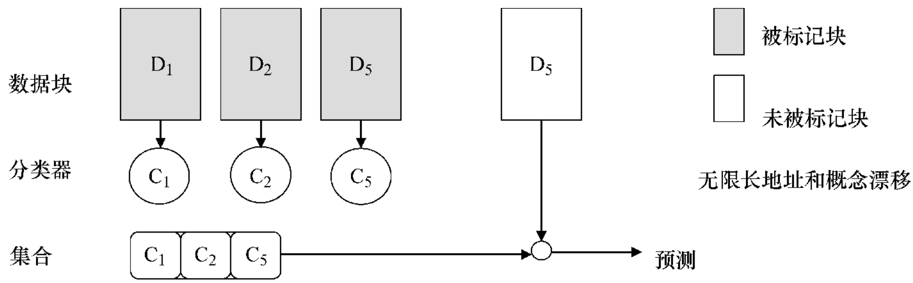


图 25. 2　集成结构

算法 1：更新分类器集成

输入：$\{D_{n-r+1}, \ldots, D_n\}$ r**个最近标记的数据块**

A：当前最好的kv个分类器集

输出：更新集合A

1: **for** each classifier $A_i \in A$ **do**

2: $e(A_i) \leftarrow$ *error of* A_i on D_n // test and compute error

3: **end for**

4: $D \leftarrow \cup_{j=n-r+1}^{n} D_j$

5: Partition D into equal parts $\{d_1, d_2, \ldots, d_v\}$

6: **for** $j = 1$ to v **do**

7: $A_j^n \leftarrow n$ newly trained classifier from data $D - d_j$

8: $e(A_j) \leftarrow$ error of A_j on d_j // test and compute error

9: **end for**

10: $A \leftarrow$ best Kv from $A^n \cup A$ *based on computed error* $e(.)$

25. 4. 2　误差减少分析

正如在算法 1 中解释的，我们构建了 Kv 个分类器的集合 A，使用集合中分类器多数投票方法来对测试实例 x 分类。我们使用简单多数表决方式而不使用加权多数表决（参见本章参考文献［WANG03］）。因为，对数据流来说，简单多数表决已经从理论上被证明是最优选择（见本章参考文献［GAO07］）。加权表决在这些情况下可能会出问题，因为它假设训练和测试数据的分布是相同的。然而，在数据流中，由于概念漂移，这个假设是不成立的。因此简单多数表决是一个更好的选择。我们在实验中也证明了这一观点，简单多数表决比加权多数表决的结果要好。

我们已经在本章参考文献［MASU11］中显示，与单分区单块（SPC）方法相比，EMPC 可以进一步减少对概念漂移数据流进行分类的预期误差，单分区单块（SPC）方法仅使用一个数据块来训练单个分类器（即 $r=v=1$）。更直观地看出，减少误差有两

个主要原因。第一，通过引入多块概念，增加每个分类器的训练数据。较大的训练数据自然会导致更好的训练模型，从而减少误差。第二，相对于从训练数据中训练一个模型，我们把数据分成 v 个分区，并且为每个分区训练一个模型。这将进一步降低误差，因为 v 个分类器集成整体误差的平均期望理论上为单个分类器低 $1/v$（见本章参考文献［TUME96］）。因此，无论多块还是多分区策略都有利于降低误差。

25.4.3　经验性误差减少与时间复杂度

对于给定的分区大小 v，增加窗口大小 r 只会在一定程度上减少误差。然而，增加 r 实际上损害了算法的性能。另外，r 的上界取决于飘移 ρd 的大小。我们在本章参考文献［MASU11］中显示了 $v=2$ 的相对误差 ER（Relative Error），由于 r 的增加，ρd 具有不同的值。我从图中可以明显看出，降低ρd的值，增加 r，可以很大程度降低相对误差。但是，在所有情况下，当 r 超过一定的阈值，ER 开始大于 1。虽然不太可能从数据中确定ρd的准确值，但是我们可以通过实验确定 r 的最优值。在实验中，我们发现对于较小块数时，r 值越高越好，反之亦然。但是，最好的性价比权衡是 $r=2$ 或 $r=3$。我们在实验中已经使用 $r=2$。同样，对于一个固定值 r，可以确定 v 的上限。应该注意的是，如果 v 增加，运行时间也随着增加。在实验中，我们获得了最好的性价比权衡是在 $v=5$ 时。

算法的复杂度为 $O(vn(Ks+f(rs)))$，n 是数据块的总数，s 是每个块的大小，$f(z)$ 是在训练数据大小为 z 的情况下建立一个分类器所需要的时间。v 是常数，复杂度为 $O(n(Ks+f(rs)))$。

25.4.4　Hadoop/Map/Reduce 框架

我们使用开源的 Hadoop/Map/Reduce 框架来实现我们的实验（见本章参考文献［APAC10］）。在这里我们提供了 Hadoop/Map/Reduce 框架特征提取和选择算法的一些算法细节。Map/Reduce 框架中的 Map 函数将一个键－值对作为输入，并为每个键－值对提供一列中间键－值对。

Map：(MKey × MVal) → (RKey × RVal) *

所有的 Map 任务都由集群中的每个节点并行处理，而不与其他节点共享数据。Hadoop 通过对共享公共中间键 k ∈ RKey 的每组中间值 V ⊆ RVal 进行分组，来核对 Map 任务的输出。由此产生核对后的（k，V）作为输入传入 Reduce 节点。因此，Hadoop/Map/Reduce 框架中的每个规约器都会以迭代的方式，一次收到 Hadoop 发布的多个（k，V）对的列表。所以 Reduce 可以理解为具有签名功能。

Reduce：(RKey × RVal *) * → Val

上面的 Val 是 Map/Reduce 循环最后结果的类型。

在我们的框架中，Map 键（MKey）是二进制文件标识符（例如，文件名），Map 值（MVal）是以字节为单位的文件内容。Reduce 键（RKey）是 n 元语法特征，对应的值（RVal）是发现它们所处位置的文件实例的类别标签。算法 2 显示了 Map 节点用于将前者映射到后者的特征提取过程。

算法 3 的第 5～10 行总结了 Map 报告的类别标签，以获得每个 n 元语法的正确和错误的实例计数。这为第 11 行计算每个 n 元语法的信息增益奠定了基础。第 12～16 行使用最小堆数据结构 h 过滤除了被信息增益评估的 S 个特征。最终遇到的所有 S 个的最佳特征被第 18～20 行返回。

因此 Hadoop 系统中的规约器 q 产生 qS 个候选特征及其信息增益。这些数据流被送到第二个规约器，算法 3 的后半部分进行了简单实现，目的是来选择最好的 S 个特征。

算法 2: *Map(file_id, bytes)*

输入：文件，带有字节内容的文件标识。

输出：(g, l)**对，这里** g **是** n-gram，l **是文件标识的标签**

1. $T \leftarrow \varnothing$
2. **for all** n-grams g in bytes **do**
3. $T \leftarrow T \cup \{g, labelof(file\ id)\}$ $\{(g, labelof(fil_id))\}$
4. **end for**
5. **for all** $(g, l) \in T$ **do**
6. **print** (g, l)
7. **end for**

算法 3: $Reduce_{p,t}(F)$

输入：(g, L)**对的F列表，**g **是**n-gram，L**是类标签的列表；**t**是原始实例集合的点大小** p**是正面实例总个数**

输出：S **个** (g, i) **对，其中**i**是**g**的信息增益**

1. **heap** h/* empty min-heap */
2. **for all** (g, L) in **F do**
3. $t' \leftarrow 0$
4. $p' \leftarrow 0$
5. **for all** l **in** L **do**
6. $t' \leftarrow t' + 1$
7. **if** $l = +$ **then**
8. $p \leftarrow p + 1$
9. **end if**
10. **end for**
11. $i \leftarrow \hat{G}\left(p', t', p, t\right)$
12. **if** $h.\ size < S$ **then**
13. $h.\ insert(i_{(g)})$
14. **else if** $(h.\ root < i)$ **then**
15. $h.\ replace(h.\ root, i_{(g)})$
16. **end if**
17. **end for**
18. **for all** $i_{(g)}$ in h do
19. print (g, i)
20. **end for**

25.5　恶意代码检测

25.5.1　简介

恶意软件是网络攻击的主要来源。一些恶意软件的种类是纯静态的，每个实例是传播它的精确副本。一旦识别出单个实例，这些副本相对容易被检测和过滤。然而，当前恶意软件的一个更明显的特征是多态的。多态恶意软件在传播过程中自我修改，以便每个实例都具有不同的语法，但却携带语义相同的恶意载荷。软件杀毒界投入大量的努力和人力设计、部署自动化算法，来检测特定的恶意软件实例和那些已经被人类专家确定并分析的多态恶意软件家族。这导致恶意软件作者和杀毒专家之间的竞赛将持续升级，每个阵营都试图发展防御措施和对抗对方最近的进展。随着恶意软件开发的不断增加和恶意软件变种的指数增长，许多人认为这种竞争最终对于防御者来说是一场失败的战斗。

恶意代码检测问题，可以建模为无限长和概念漂移数据流的数据挖掘问题。概念漂移随着多态恶意软件变种而出现，攻击者和防御者都将引进新的技术用于开展竞赛。这个概念的实用化促使了我们的流分类技术的应用，从而自动检测新的恶意可执行文件。

使用 n – grams 进行特征提取涉及从给定的数据集（训练数据）中提取所有可能的 n – grams，并且需要挑选其中最佳的 n – grams。每一个这样的 n – grams 就是一个特征。即，一个 n – grams 是一个 n 字节序列。在提取 n – grams 之前，我们进行预处理，将二进制可执行文件转换成十六进制文件。此时，粒度级别是 1 字节。我们使用 UNIX 十六进制转换工具把二进制可读文件转换成文本文件（十六进制文件），其含有对应每个二进制字节的十六进制数。执行此过程来确保二进制可执行文件的安全和便捷的可移植性。在非分布式框架中，特征的提取过程包括两个阶段（简短地说是）：特征选择和特征提取。本章介绍我们这种传统技术在云计算中的变体方式。

25.5.2　非分布式特征提取与选择

在非分布式环境下，特征提取过程如下。用一个 n 字节的滑动窗口扫描每个十六进制文件的内容。每一组出现在窗口中的 N 字节序列是一个 n – grams。对每个 n – grams，当 g 出现时，我们统计到文件实例的总数 tg 中，当出现总数 $pg \leqslant tg$ 时就呈现为正值（例如恶意可执行文件）。

这涉及维护到目前为止遇到的所有 n – grams 的哈希表 T。如果在 T 中没有找到 g，则根据当前文件是否具有负或正类标签，将 g 加到 T 中，其中 tg ＝ 1 和 pg ∈ ｛0，1｝。如果 g 已经存在于 T 中，tg 要递增并且 pg 根据文件的标签也有条件地递增。当扫描完所有的十六进制文件，T 包含数据集中所有唯一的 n – grams，以及它们在正实例和总体的频率。

从对应当前块的所有文件中提取出的 n – grams 特征，并不总是全部都实用。指数级数量的这些 n – grams 可能导致不可接受的内存开销，会使训练过程变慢，或者使分类器混淆，产生大量噪声、冗余和不相关特征。为了避免这些陷阱，候选 n – grams 特

征必须根据一个选择准则进行排序，这样只有最优的 n – grams 会被选择。

我们选择信息增益作为挑选标准，因为在相关文献中这是挑选特征最有效的方法之一。信息增益可以定义为分类训练数据时属性（即特征）的有效性度量。如果我们根据属性值分解训练数据，分解完成后，信息增益测量熵的期望减少。训练数据中属性的熵减少越多，这个属性越适合用于分类数据。

我们在本章参考文献［MASU11］中已经说明，当新的特征被考虑时，它们的信息增益会和堆的根对比。如果新特征的信息增益比根多，这个根就被去除，并且新的特征将加入堆中，否则，去除新的特征，特征选择继续进行。

25.5.3 分布式特征提取与选择

这里会存在与刚才描述的非分布特征提取和选择方法有关的几个缺陷。

■ 提取的 n – grams 特征的总数可能会非常大。例如，一个数据块中的 4 – grams 个数大概是 2 亿。在主内存中存储它们几乎是不可能的。一个显而易见的解决方案是把它们存储在一个硬盘文件中，但这会导致无法接受磁盘读写操作的开销成本。

■ 如果冲突特征在哈希表 T 中不排序，每个被扫描过的 n – grams 在特征提取时均需要被线性搜索，来确定是否存在于哈希表 T 中。如果它们已经被归类，那么插入时需要线性扫描。在任一情况下，提取所有 n – grams 的最长时间是每个块中 n – grams 总数 N 的二次方，当 $N \approx 108$ 时所花费的时间是不切实际的。同样，非分布式特征选择过程需要每个块中的一种 n – grams。在一般情况下，这需要花费时间复杂度为 O（$N \log N$），当 N 很大时，时间开销是不切实际的。

为了高效快捷地解决非分布式特征提取与选择的缺点，我们利用云计算的力量。利用 Hadoop/Map/Reduce 框架并行处理特征提取、n – grams 排序与特征选择。

Map/Reduce（见本章参考文献［DEAN08］）是云计算环境中一种日趋流行的分布式编程范例。该模型能并行处理大型数据集，以多种方式将工作负载分布在许多节点（机器）上，主要关注的是简化大型数据集使用廉价集群计算机的过程。另一个目标是易于使用的负载平衡和容错能力。

Map/Reduce 以它的两个主要功能命名。Map 功能将任务分解成子任务，并分配到可用节点上。而它的第二个功能 Reduce，则是聚集已经完成子任务的结果。我们今后将执行这些功能的节点分别称为映射器和规约器，Map/Reduce 对 n – gram 特征提取和选择的过程将在附录中给出解释，在本节中，我们给出这种方法的更详细的概述。

用于提取 n – grams 每个训练块包含 N 个训练文件。这些训练文件首先根据 HDFS（图 25.3，步骤 1）分配到 m 个节点（机器）。节点数量 m 根据系统适用性由 HDFS 选择。每个节点然后独立地从节点的训练文件子集中提取 n – grams，训练文件使用技术（见本章参考文献［MASU11］）（图 25.3，步骤 2）提供给节点。当所有节点完成其作业时，将从每个节点中提取的 n – grams 进行整理（图 25.3，步骤 3）。

例如，假设节点 1 在一个正实例下观察到 n – grams abc（即，恶意训练文件），而节点 2 在负（即良性的）实例中也观察到它。在图 25.3 中，节点 1、2 上分别标记几对 abc，+和 abc，–。当 n – grams 组合时，包含相同 n – grams 实例的标签被聚合。因此，

abc 的聚合对是 abc，+ −。这些组合的 n − grams 被分配到 q 个规约器（q 基于系统适用性由 HDFS 选择）。每个规约器，首先计算集合标签以获得正计数和总计数。在 n − grams 的 abc 例子中，我们得到标签是 $p_{abc}=1$ 和 $t_{abc}=2$。规约器使用这些标签从提供给节点（图 25.3，步骤 5）的所有 n − grams 的子集中选择最佳的 S 个 n − grams。这样就可以使用一个大小为 S 的最小堆来有效完成。这个过程需要的时间复杂度为 O（$W\log S$），W 是提供给每个规约器 n − grams 的总数。相反，非分布式版本需要的时间复杂度为 O（$W\log W$）。因此，从 q 个规约器节点中，我们就可以获取 qS 个 n − grams。类似上述过程，我们再次通过运行另一轮 Map/Reduce 循环选择最佳 S 个 n − grams，这一循环中映射阶段为空，但是规约阶段仅使用一个节点就可以执行特征选择（图 25.3，步骤 6）。特征集中的每个特征是二进制，如果在给定实例中，它的值就为 1（即可执行文件）否则为 0。对于每个训练或者测试实例，我们要计算特征向量，其中的每一个位由对应特征集的特征值构成。这些特征向量被分类器用来训练和测试。

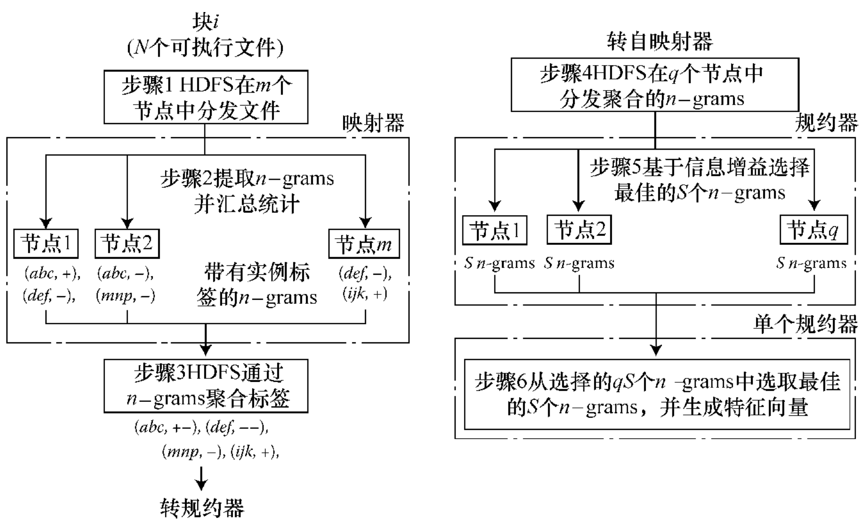


图 25.3 分布式特征提取和选择

25.6 实验

我们基于合成数据、可控环境下的僵尸网络和恶意软件数据集评估我们的方法，并把实验结果与几种基线方法相比较。

25.6.1 数据集

25.6.1.1 合成数据集

为了产生概念偏移的合成数据，我们使用由 $\sum_{i=1}^{d} a_i x_i = a_0$ 给定（见本章参考文献［WANG03］）移动的一个超平面。如果 $\sum_{i=1}^{d} a_i x_i \leqslant a_0$，则这个示例是负的，否则是正的。每个示例是一个随机生成的 d 维向量 $\{x_1,\cdots,x_d\}$ $x_i \in [0,1]$，8 个 $\{a_1,\cdots,a_d\}$ 也

是在［0，1］之间随机初始化的真实数据，a_0 的值可以被调整，以便产生相同数量数据的正和负示例。这可以通过选择 $a_0 = (1/2)\sum_{i=1}^{d} a_i$ 来完成。在我们的实验中 $p = 5n$ 时，通过切换示例标签的百分比来引入随机干扰。有许多参数可以模拟概念偏移。这里，使用本章参考文献［WANG03］中相同的参数。总的来说，我们产生的 25 万个记录和 4 个不同的数据集包含块的大小分别为：250、500、750 和 1000。每个数据集含有各 50% 的正实例和负实例。

25.6.1.2 僵尸网络数据集

僵尸网络由互联网中被攻破的主机组成，所以控制权在人类攻击者之中，攻击者被称为僵尸主控者（见本章参考文献［BARF06］）。僵尸主控者发出命令，僵尸机器执行恶意行为，比如发起 DDoS 攻击、垃圾邮件、间谍操作等。僵尸网络成为目前互联网发展的严重威胁。随着僵尸网络拓扑结构演变，许多前沿僵尸网络使用对等（Peer-to-Peer，P2P）技术来进行可靠和隐蔽的通信。这些僵尸网络是分布式和小型的，使得它们更难以被检测和销毁。P2P 僵尸网络的例子包括 Nugache（见本章参考文献［LEMO06］）、Sinit（见本章参考文献［STEW03］）和 Trojan. Peacomm（见本章参考文献［GRIZ07］）。

僵尸网络流量可以看成具有无限长和概念偏移的数据流。概念偏移发生在机器执行新的恶意任务时，或者采用不同的沟通策略，以响应新的主控机者指令。因此我们认为数据流分类技术非常适合检测 P2P 僵尸网络流量。

我们在可控环境下使用 P2P Nugache 僵尸主机（见本章参考文献［LEMO06］）来产生真实 P2P 僵尸网络流量。特征提取过程的细节在 Masud 等人的文章（见本章参考文献［MASU08b］）中讨论。一共有 81 个连续属性，整个数据集包含 3 万个记录，代表了一个星期的网络流量。我们产生 4 个不同的数据集，包含的数据块大小分别为 30、60、90 和 120 分钟。每个数据集还有 25% 的正实例（僵尸网络流量）和 75% 的负实例（良性流量）。

25.6.1.3 恶意软件数据集

我们从不同的 Windows 计算机中提取一共 38694 个良性可执行文件，并从在线恶意软件仓库 VX Heavens（见本章参考文献［VX10］）中收集到的 66694 个恶意可执行文件，其中包含大量的恶性可执行文件（病毒、蠕虫、木马、后门）。良性的可执行文件包括在 Windows 安装文件夹中发现的各种应用程序，以及默认程序安装目录中的其他可执行文件。

在这两种情况下，我们只选择 Win32 的可移植的可执行文件（Portable Executables，PE），使用 ELF 可执行程序进行实验是今后工作的一个潜在方向。收集的 105388 个文件（良性和恶意）形成了 130 个块的数据流，每个数据流包含 2000 个实例（可执行的小程序）。通过对恶意软件的版本和发现日期进行排序，模拟网络恶意软件的演变性。每个块有 1500 个良性可执行文件（75% 为负数）和 500 个恶意可执行文件（25% 为正）。关于其中的特征提取和选择过程，在本章以前的部分已经描述过。

应注意到所有这些数据集在本质上都是动态的。它们无界限限制（可能无限长）使得它们超出了纯粹静态分类框架的范围。另外，合成的数据也表现出概念偏移。虽然很难准确判断数据集中是否还有概念偏移，经过长时间观察，理论上可执行文件的数据流会有概念偏移。在每个块中识别的不同特征集，可以证明恶意软件数据能够揭示特征演变。

25.6.2 基线方法

为了分类，我们使用 Weka 机器学习开源包（见本章参考文献［HALL09］）。我们使用两种不同的分类器：J48 决策树和 Ripper。然后，我们将下面的每个基线技术与我们的 EMPC 算法进行比较。

BestK。这是一个 SPC 集成方法，它使用一组最佳 K 个分类器。通过存储所有已知的分类器，在最近训练数据块上根据期望误差来挑选最佳 K 个分类器。对于实例，使用简单多数表决方式来测试。

Last。这种情况下，我们只保留在最近训练块上训练的分类器，这可以认为是一个 $K=1$ 情况下的 SPC 方法。

AWE。这是使用精度加权分类器集成，来实现 SPC 的方法（见本章参考文献［WANG03］）。它建立一个 K 个模型集合，每个模型从一个数据块中被训练。集合更新如下，使用 Cn 表示最近训练数据块上的分类器，从已存在的 K 个模型和最新模型 Cn 中，基于在最近数据块上的误差挑选最佳 K 个模型。挑选时基于加权投票，每个模型的权重和在最近训练块上模型得到的误差成反比。

ALL。SPC 使用所有已知的分类器进行集成。新的数据块使用该集成，通过简单表决方式来测试。既然这是一个 SPC 方法，每个分类器就只被一个数据块训练。

对于大多数数据集，我们获得 r 和 v 的最佳值分别是 2 ~ 3 之间和 3 ~ 5 之间。除非有其他情况，在我们的实验中使用 $r=2$ 和 $v=5$。为了获得相对公平的比较，我们在 EMPC 和所有基线技术中均使用了相同的 K（集合大小）值。

25.6.2.1 Hadoop 分布式系统配置

我们进行实验的分布式系统由一个 10 个节点的集群组成。每个节点都由相同的硬件组成：Intel Pentium IV 2.8GHz 处理器、4GB 的主内存和 640GB 的硬盘空间。其软件环境包括一个 Ubuntu 9.10 操作系统、Hadoop - 0.20.1 分布式计算平台、JDK 1.6 Java 开发平台，以及 100Mbit/s 的 LAN。

25.7 讨论

我们的工作考虑了一个由纯语法特征组成的特性空间：包括从可执行代码段获取的二进制 n - grams、静态数据段、首部和不可信文件的所有其他内容等。高级的结构特性，比如 Calland 控制流图；动态特性，比如运行时跟踪，超出了我们目前研究的范围。然而，正如以前的大量工作还有我们的实验所证明的那样，大家已经观察到 n - grams 特征对于恶意软件检测具有非常高的辨别力。这其中部分是因为 n - grams 集包含了整个二进制内容，包括头和数据表，还捕获了一些重要的底层结构细节，这些细节常常被高层次抽象的概念表示出来。例如，恶意软件通常包含已汇编和使用非标准工具链接的手工汇编代码。这将允许攻击者实现二进制的混淆和低级别的使用，而不是来自更高级别的源语言和标准编译器。带来的结果就是，恶意软件经常包含特殊指令的编码结构、

标题结构，和只能被以二进制形式看到的异常的链接表，这些指令不在汇编代码列表中，而在控制流程图或系统 API 调用痕迹中。如果扩展特征空间来包含这些额外的更高等级的特征，需要一个高效和可靠的方法在特征选择时获取和评估他们相关鉴别能力，这将作为未来工作的主题。

报告本章参考文献［MASU11］中报道的实证结果证实了我们的分析：多分区、多块方法比单分区、单分块方法表现得更好。直观地来说，一个分类器在多块上的预测准确度要比在单分块上训练得要好，因为训练数据更大。此外，如果集成分类器被多分区训练数据所训练，预测准确度应该比单个分类器在同样的训练数据上的要好，因为集成分类器的降低误差能力要比单个分类器高。此外，EMPC 的精度优势可以追溯到两个重要的差异，这两个差异存在于我们的工作和 AWE 之间。第一，在 AWE 中，当一个分类器在集成更新中被移除时，所有从相关块中获得的信息被遗弃。但是在 EMPC 中，一个或更多早期块的分类器可能会存活下来。因此，EMPC 集成更新比 AWE 趋于保留更多的信息，从而生成更好的集合。第二，AWE 需要至少 Kv 个数据块，而 EMPC 需要至少 $K+r-1$ 个数据块来得到 Kv 分类器。因此，与 EMPC 相比，AWE 倾向于保留更过时的分类器，这导致一些过时的分类器对分类准确度产生负面影响。

但是，得到更高准确度的同时会带来更高的运行时间成本。理论上，EMPC 最多比 AWE 慢 rv 时间，而在精度上却较为接近。这在实验评估上也是显而易见的，它显示 EMPC 的运行时间是 AWE 的 5 倍之内（$r=2$，$v=5$）。但是，可以采取一些优化措施来降低运行时间损耗。首先，可以很容易地实现对每个分区训练的并行化，通过因子 v 来减少训练时间。第二，在整体中每个模型的分类器也可以并行进行，从而通过因子 Kv 减少了分类时间。因此，至少通过因子 v 使训练和分类的并行化来减少运行时间，这使得时间花费和 AWE 接近。或者，如果并行化是不可用的，则可以将参数 v 和 r 降低到牺牲预测精度，以降低运行时间成本。在这种情况下，获取运行时间和预测准确度之间的期望的平衡，就要评估数据流最开始的几个块所使用的不同的 v 和 r 值，并选择最合适的值。

25.8 总结和展望

许多入侵检测问题可以被归类为无限长、概念漂移数据流的分类问题。概念漂移往往出现在攻击者回应和采取防御时的数据流中。我们明确阐述了恶意代码检测和僵尸网络流量检测等问题，并引入了 EMPC。它是一种全新的集成学习技术，用于无限长度、概念漂移数据流的自动分类。通过应用 EMPC 方法，我们从动态恶意软件和僵尸网络流量样本中，获得的真实数据流比其他数据流分类技术拥有更好的检测准确度。表明这种方法对于入侵检测和一般数据流分类是有用和高效的。

EMPC 使用广义的、多分区、多块集成学习技术。理论和实验评估显示，这项技术与已知的单分区、单块集成方法相比，可以显著减低期望分类误差。此外，我们还表明，在基于 Map/Reduce 的云计算框架中（见本章参考文献［DEAN08］），EMPC 可以很好地实

现，其结果是一个低成本、可伸缩的流分类框架，具有较高的分类精度和低运行开销。

我们在未来的工作方向上，至少有两个技术扩展可以去做。第一，我们在目前的特征选择程序集中在基于信息增益的最优 S 特征，并以此作为选择标准。通过利用近年来的监督维数减少技术（见本章参考文献［RISH08］、［SAJA05］）改进特征选择，可以提高分类精度。第二，我们方法的运行时间可以通过利用额外的云计算并行架构来改善。比如，一个集成的分类器可以作为 MapReduce 框架中的映射器并行运行，而规约器收集的结果用来进行投票。类似地，下一个整合的候选分类器可以同时进行训练和评估。按照此方法重新构造系统的集成组件，可以很大程度缩短处理时间，因此，有机会将更多的处理时间用于分类以便提高准确性。

参考文献

[AGGR06] Aggarwal, C.C., J. Han, J. Wang, and P. S. Yu, A framework for on-demand classification of evolving data streams. *IEEE Trans. Knowl. Data Eng.* 18(5): 577–589, 2006.

[AHA91] Aha, D.W., D. Kibler, and M.K. Albert, Instance-based learning algorithms. *Mach. Learn.* 6: 37–66, 1991.

[APAC10] Hadoop. hadoop.apache.org., 2010.

[BARF06] Barford, P. and V. Yegneswaran, An inside look at botnets. In *Malware Detection, Advances in Information Security* (Editors: M. Christodorescu, S. Jha, D. Maughan, D. Song, and C. Wang,), Springer, 171–192. 2006.

[BIFF09] Bifet, A., G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, Paris, France, 139–148, 2009.

[BOSE92] Boser, B.E., I.M. Guyon, and V.N. Vapnik, A training algorithm for optimal margin classifiers. In *Proceedings of the 5th ACM Workshop on Computational Learning Theory*. 144–152, Pittsburgh, PA, 1992.

[CHEN08] Chen, S., H. Wang, S. Zhou, and P.S. Yu, Stop chasing trends: Discovering high order models in evolving data. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*. 923–932, 2008.

[COHE96] Cohen, W.W., Learning rules that classify e-mail. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, Palo Alto, CA, 18–25, 1996.

[COMP07] Computer Economics, Inc. Malware report: The economic impact of viruses, spyware, adware, botnets, and other malicious code. http://www.computereconomics.com/article.cfm?id=1225, 2007.

[CRAN05] Crandall, J. R., Z. Su, S. F. Wu, and F.T. Chong, On deriving unknown vulnerabilities from zero-day polymorphic and metamorphic worm exploits. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'05)*. 235–248, 2005.

[DEAN08] Dean, J. and S. Ghemawat, MapReduce: Simplified data processing on large clusters. *Commun. ACM* 51(1): 107–113, 2008.

[DOMI00] Domingos, P. and G. Hulten, Mining high-speed data streams. In *Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, Boston, MA, 71–80, 2000.

[FAN04] Fan, W. Systematic data selection to mine concept-drifting data streams. In *Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. 128–137, Seattle, WA, 2004.

[FREU96] Freund, Y. and R.E. Schapire, Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*. 148–156, Bari, Italy, 1996.

[GAO07] Gao, J., W. Fan, and J. Han, On appropriate assumptions to mine data streams: Analysis and practice. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*. 143–152, Omaha, NE, 2007.

[GRIZ07] Grizzard, J. B., V. Sharma, C. Nunnery, B.B. Kang, and D. Dagon, Peer-to-peer botnets: Overview and case study. In *Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots)*. 1–8, Cambridge, MA, 2007.

[HALL09] Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, The WEKA data mining software: An update. *ACM SIGKDD Explor.* 11 (1): 10–18, 2009.

[HAML09] Hamlen, K.W., V. Mohan, M.M. Masud, L. Khan, and B.M. Thuraisingham., Exploiting an antivirus interface. *Comput. Stand. Interfaces* 31(6): 1182–1189, 2009.

[HASH09] Hashemi, S., Yang, Y., Mirzamomen, Z., and Kangavari, M. R. Adapted one-versus-all decision trees for data stream classification. *IEEE Trans. Knowl. Data Engin.* 21(5):624–637, 2009.

[HULT01] Hulten, G., L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. 97–106, San Francisco, CA, 2001.

[KOLT04] Kolter, J. and M.A. Maloof, Learning to detect malicious executables in the wild. In *Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. Seattle, WA, 470–478, 2004.

[KOLT05] Kolter, J. Z. and M.A. Maloof, Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*. Bonn, Germany, 449–456, 2005.

[LEMO06] Lemos, R., Bot software looks to improve peerage. *Security Focus*. http://www.securityfocus.com/news/11390.

[LI06] Li, Z., Sanghi, M., Y. Chen, M.-Y. Kao, and B. Chavez, Hamsa: Fast signature generation for zero-day polymorphic worms with provable attack resilience. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. Berkeley, CA, 32–47, 2006.

[MASU08a] Masud, M. M., J. Gao, L. Khan, J. Han, and B. Thuraisingham, Mining concept-drifting data stream to detect peer to peer botnet traffic. Tech. rep. UTDCS-05-08, The University of Texas at Dallas, Richardson, Texas. http://www.utdallas.edu/~bxt043000/Publications/Technical-Reports/UTDCS-05-08.pdf, 2008a.

[MASU08b] Masud, M.M., L. Khan, and B. Thuraisingham., A scalable multi-level feature extraction technique to detect malicious executables. *Inf. Syst. Frontiers,* 10(1):33–45, 2008b.

[MASU09] Masud, M. M., J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, 2009. A multi-partition multi-chunk ensemble technique to classify concept-drifting data streams. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. Bangkok, Thailand.

[MASU11] Masud, M. M.,T. Al-Khateeb, K. W. Hamlen, J. Gao, L. Khan, J. Han, B.M. Thuraisingham, Cloud-based malware detection for evolving data streams. *ACM Trans. Management Inf. Syst.* 2(3): 16, 2011.

[MICH94] Michie, D., D. J. Spiegelhalter, and C. C. Taylor, Eds. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Series in Artificial Intelligence. Morgan Kaufmann, 50–83, 1994.

[NEWS05] Newsome, J., B. Karp, and D. Song, Polygraph: Automatically generating signatures for polymorphic worms. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 226–241, 2005.

[QUIN03] Quinlan, J. R., *C4.5: Programs for Machine Learning* 5th Ed. Morgan Kaufmann, San Francisco, CA.

[RISH08] Rish, I., Grabarnik, G., Cecchi, G. A., Pereira, Gordon, G., Closed-form supervised dimensionality reduction with generalized linear models. In *Proceeding of Intl. Conf. on Maching Learning* (*ICML 2008*), Helsinki, Finland, 832–839, 2008.

[SAJA05] Sajama and A. Orlitsky, Supervised dimensionality reduction using mixture models. In *Proceedings of the 22nd ACM International Conference on Machine Learning (ICML)*. 768–775, Bonn, Germany, 2005.

[SCHO05] Scholz, M. and R. Klinkenberg, An ensemble classifier for drifting concepts. In *Proceedings of the 2nd International Workshop on Knowledge Discovery in Data Streams (IWKDDS)*. 53–64, 2005.

[SCHU01] Schultz, M. G., E. Eskin, E. Zadok, and S. J. Stolfo, Data mining methods for detection of new malicious executables. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 38–49, 2001.

[STEW03] Stewart, J. Sinit P2P trojan analysis. http://www.securiteam.com/securityreviews/6J0022A95E.html, 2003.

[TUME96] Tumer, K. and J. Ghosh, Error correlation and error reduction in ensemble classifiers. *Connect. Sci.* 8(3):385–404, 1996.

[VX10] VX Heavens 2010. VX Heavens. vx.netlux.org.

[WANG03] Wang, H., W. Fan, P. S. Yu, and J. Han, Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. Washington, DC, 226–235, 2003.

[YANG05] Yang, Y., X. Wu, and X. Zhu, Combining proactive and reactive predictions for data streams. In *Proceedings of the 11th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. Chicago, IL, 710–715, 2005.

[ZHAN09] Zhang, P., X. Zhu, and L. Guo, Mining data streams with labeled and unlabeled training examples. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM)*. Miami, FL, 627–636, 2009.

[ZHAO09] Zhao, W., H. Ma, and Q. He, Parallel *K*-means clustering based on MapReduce. In *Proceedings of the 1st International Conference on Cloud Computing (CloudCom)*. Beijing, China, 674–679, 2009.

第 26 章　用于内部威胁检测的基于云计算的数据挖掘

26.1　概述

对内部威胁的有效检测，需要比外部威胁的检测要有更精细的监视机制。这些监视器必须有效和可靠的部署在软件环境中，并能够及时捕获内部恶意行为。这样的环境通常包含用户级别的应用程序，比如文字处理器、电子邮件客户端和 Web 浏览器，对于此类程序来说通过传统的方法来进行可靠的监视内部活动是困难的。

为了监视内部活动，监视工具需要捕获这些内部软件之间的通信和关系，并存储、查询这些关系，并且最终通过分析这些关系来提取关系模式，使得分析师可以更好地洞察潜在的威胁。随着时间的推移，内部程序之间的通信与关系的数量可以达到数十亿。使用我们项目中开发的工具，可以使得数十亿内部程序之间的关系被捕获、存储、查询和分析，并以此来检测恶意内部威胁程序。

在本章中，我们将讨论数据挖掘技术如何在云计算环境下，被运用到内部程序威胁检测中。首先，我们将讨论语义 Web 技术如何被运用到表示内部程序之间的通信。其次，将讨论我们的内部程序检测方法。最终，提供一个我们内部威胁检测框架的概述，也包含一些其他技术。

本章结构如下。26.2 节，我们将讨论面临的挑战、相关工作，以及对与此问题我们的解决方法。我们的方法将在 26.3 节中详细讨论。我们的框架将在 26.4 节讨论。本章总结将在 26.5 节。图 26.1 说明了本章包含的内容。

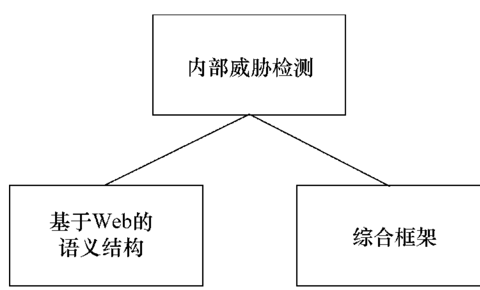


图 26.1　用于内部威胁检测的基于云计算的数据挖掘技术

26.2　挑战、相关工作和我们的方法

内部程序和内部程序之间的关系如同图形中的节点和链接。因此，面临的难点是在图形中表示信息，制定有效的存储策略，开发查询处理图形的技术，并随后开发从图形中提取信息的数据挖掘和分析技术。特别的是，有三大挑战：

1）以一种具有表达性和统一的方式，把这些大型图形存储在辅助存储器中。

2）设计可扩展的解决方案，查询这些大型图形并发现相关数据。

3）识别复杂图形中的相关特征，并在动态环境中检测随时间变化的内部程序威胁。

这个方法的动机是解决之前提出的三个挑战，我们基于云计算开发了解决方案，①描述包含数十亿节点和节点之间的边，所代表的活动（例如，信用卡交易、电子邮件或者短信）。由于图形是巨大的，我们将开发高效和持久的存储技术。②为了方便新型异常检测，我们需要一个高效的接口，及时从这种持久存储中提取相关数据。因此，我们将在存储的图形中开发有效的查询技术。③所提取的相关数据可用于进一步分析，以检测异常。为此，首先，我们必须从复杂图形中识别相关的特征，然后开发挖掘大型图形，以提取挖掘的技术。

正如第一章所述，内部威胁检测是一个困难的问题（见本章参考文献［MAYB05］)。随着越来越多来自不同来源和传感器的数据，这个问题就变得越来越复杂。最近有一些研究工作重点关注根据图表进行基于异常的内部威胁检测（见本章参考文献［EBER09］)。这种方法基于最小描述长度准则（Minimum Description Length，MDL)。在本章参考文献［EBER09］的讨论中，指出它具有一些局限性。首先，对于这个方法，扩展性是一个问题。换句话说，他们没有讨论任何关于大图形的问题。其次，异质性的问题没有得到解决。最后，目前还不清楚他们的算法如何处理随时间变化的动态环境。

如今已经开发了几种图形挖掘技术，尤其对社会网络分析（见本章参考文献［COOK06］、［TONG09］、［CARM09］和［THUR09］)。但这些技术的可扩展性始终是一个问题。一些从事数学研究的社团，他们的工作是应用线性规划技术来进行图形分析（见本章参考文献［BERR07］)。这些技术是否对真实环境有效还是不清楚的。

对于一个可行的解决方案，必须具有高度可扩展性，并且支持多异构数据源。目前先进的解决方案扩展性并不好，且准确度也不高。通过利用 Hadoop 技术，我们的方法将是高度可扩展的。此外，利用灵活的 Web 语义 RDF 数据模型，我们可以轻易地整合排列异构数据。因此，我们的方法将创建一个在动态环境中可扩展的解决方案。目前还没有现存的威胁检测工具提供这种级别的可扩展性和互操作性。我们将使用数据挖掘技术及结合相关技术，来创建一个完整的内部威胁检测解决方案。

我们已经开发了基于 Hadoop/Map/Reduce 技术的云计算架构。内部程序与它们之间的联系是通过图的节点和连接来表示。特别的，在我们的方法中，数十亿的节点和连接通过 RDF 图表示。通过利用 RDF 表示，我们将解决异质性问题。我们将开发存储 RDF 图的有效机制，使用 SPARQL 技术查询图，并且在云计算框架中挖掘图，以提取相关模式。

26.3　用于内部威胁检测的数据挖掘

26.3.1　我们的解决方案的体系结构

图 26.2 显示了我们解决方案的体系结构。我们的解决方案将从多个数据源收集数

据，然后提取和选择特征。在对特征降维后，数据存储在我们的硬盘存储库中。数据以RDF格式存储。因此，如果数据是以其他格式存在则需要进行格式转换。RDF是Web语义的数据格式并且表示图形数据是有力的。异常预测组件将提交SPARQL到库中挑选数据，然后输出检测到的任何内部威胁。SPARQL是针对RDF数据的查询语言，它在语法上与SQL相似。每个组成部分的详细情况将在接下来的章节中描述。对于选择关系数据模型图的RDF表示，我们的目的是有效地解决异质性问题（半结构化数据模型）。对于查询，我们将利用标准查询语言SPARQL。此外，推理是我们的框架提供的一个功能。

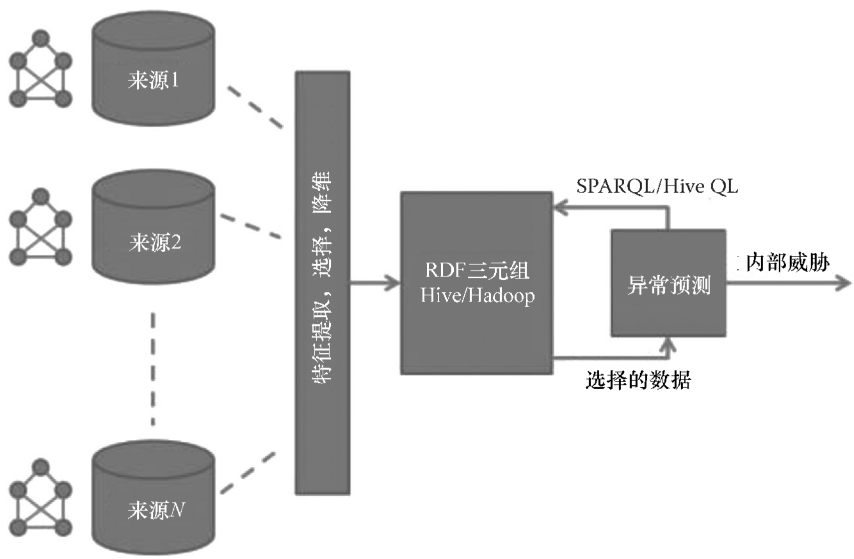


图 26.2　解决方案的体系结构

我们假设大图形已经存在。为了方便持久地存储和有效地检索这些数据，我们使用基于Hadoop云计算框架（见本章参考文献［HADO］）的分布式架构。通过利用Hadoop技术，我们的框架容错性和可扩展性较强。为了支持大量数据，我们只需简单地在Hadoop集群中加入更多的节点。集群中的所有节点都是廉价的机器，没有必要购买昂贵的服务器。为了处理大型复杂的图形，我们利用HDFS和Map/Reduce框架。前者是将数据存储在通过复制的多个节点中的存储层上。后者是可以运行Map/Reduce作业的执行层。我们使用HDFS来存储RDF数据，用Map/Reduce框架回应查询。

26.3.2　特征提取和紧凑表示

在传统的图形分析中，边是一个表示强度的简单数字。但是，我们在表示连接值时将可能面临额外的挑战，这是由于文本和电子邮件的内容都是非结构化的。一种可能的方法是将整个内容作为连接值的一部分，我们称之为显性内容（Explicit Content，EC），即使对于中等大小的图，EC也不好衡量。这是因为为了在内存中处理图形，表示两个

节点之间的连接内容需要很大的内存空间。对于每条消息，我们均使用其内容中的一个向量表示（Vector Representation of the Content，VRC）。在 RDF 三元组表示中，这将简单地表示为唯一的谓词。我们可以通过字典编码表中原始消息的物理位置或 URL，来跟踪特征向量。

VRC：预处理每个信息的过程中，我们提取关键字和短语（n-gram）作为特征。如果我们生成这些特征的向量，那么这些向量的维度会非常高。这里就是我们观察到的维度灾难（即稀疏和处理时间将增加）。因此，我们可以应用特征降维和特征选择（即主成分分析、支持向量机）。特征降维就是将高维特征空间映射到较小的空间，并且新特征维度可能是旧维度的线性组合，这个是很难解释的，因此我们需要研究特征选择。

对于特征选择，我们需要使用一组类标签来监视数据。在这里，对于信息，我们可能没有一个类标签。但是，我们知道信息的消息源/发送者和目的地/接收者。现在，我们想用这些知识来构造一个人工标签。发送方和目的地将形成一个唯一的类标签，而且所有从发送方到接收方的信息都将作为数据点。因此，我们的目标是找到适当的特征，能基于这些信息让这些类标签有辨别力。对于特征选择来说有很多方法，它们广泛地使用在机器学习的领域，比如 IG（见本章参考文献［MITC97］、［MASU10a］、［MASU10b］）、Gini 索引、卡方统计、子空间聚类（见本章参考文献［AHME09］）等。在这里我们选择信息增益，这在文本域中非常流行，选择子空间聚类来进行特征选择。

可以将 IG 定义为对特征进行分类训练数据的有效性的度量（见本章参考文献［MITC97］）。如果我们将这些属性值的训练数据分开，则 IG 就会提供分开后熵的预期减少量的测量。属性可以减少训练数据中的熵越大，分类数据的属性越好。实例 S 的集合中的属性 A 的 IG 由式（26.1）给出

$$\mathrm{Gain}(S,A) \equiv \mathrm{Entropy}(S) - \sum_{V \in Values(A)} \frac{|S_v|}{|S|}\mathrm{Entropy}(S_v) \qquad (26.1)$$

式中，值（A）是属性 A 的所有可能值的集合，S_v 是属性 A 具有值 v 的 S 的子集。S 的熵通过式（26.2）计算：

$$\mathrm{Entropy}(S) = -\sum_{i=1}^{n} p_i(S)\log_2 p_i(S) \qquad (26.2)$$

式中，$p_i(S)$ 是类 i 在集合 S 中的先验概率。

26.3.2.1 子空间聚类

子空间聚类可以用来进行特征选择。当聚类对应的数据集组成原始维度的子集时，子空间聚类是合适的。在子集形成的基础上，一个子空间聚类算法可以被使用作为软或硬子空间聚类。在软子空间聚类的情况下，根据每个特征/维度在聚类的过程中表现，对特征分配权重。然而，在硬子空间聚类的情况下，为每个聚类选择特定的特征子集，并且为该聚类丢弃其余的特征。因此，子空间聚类可以用来选择哪些特征比较重要（如果它们对所有集群的权重较小，那么就会丢弃一些特征）。一种常用的软子空间聚类方法是 SISC（见本章参考文献［AHME09］）。这种子空间聚类算法使用了下列目标函数。一个 E-M 公式用于聚类。在每次迭代中，因为要选择每个聚类具有较高权重的特

征，每个集群的特征权重实际上都要被更新。我们可以为相应数据集选择一组重要特征。

$$F(W,Z,\Lambda) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}^{f}\lambda_{li}^{q}D_{lij} * (1 + \mathrm{Im}\ pl) + \gamma\sum_{l=1}^{k}\sum_{i=1}^{m}\lambda_{li}^{q}\chi_{li}^{2}$$

这里

$$D_{lij} = (z_{li} - x_{ji})^2$$

基于以下条件

$$\sum_{l=1}^{k} w_{lj} = 1, 1 \leqslant j \leqslant n, 1 \leqslant l \leqslant k, 0 \leqslant w_{lj} \leqslant 1$$

$$\sum_{i=1}^{m} \lambda_{li} = 1, 1 \leqslant i \leqslant m, 1 \leqslant l \leqslant k, 0 \leqslant \lambda_{li} \leqslant 1$$

在这个目标函数中，W、Z、Λ 分别表示聚类成员、聚类中心和维度权重矩阵。此外，参数 f 控制每个数据点成员的模糊度，q 为进一步修改每个聚类（λ_{li}）的维度权重，并且最终由 γ 控制卡方分量和维度权重的激励强度。同时假设有 n 个文件在训练数据集中，每个特征点有 m 个特征，并且在聚类过程中有 k 个子空间聚类产生。Im pl 表示聚类杂质，χ^2 表示卡方统计。关于这些符号和聚类将如何完成的细节可以在我们之前的研究中找到（见本章参考文献［AHME09］）。应该注意的是，使用子空间聚类进行特征选择，可以被认为是一个无监督的方法，这是由于特征选择在一个无监督聚类过程中不需要任何标签信息。

一旦我们选择了特征，两个节点之间的一个消息，就会用这些特征表示为一个向量。每个向量的单个值可以是二进制或加权。因此，这将是一个原始消息的紧凑表示法，并且它可以随图形结构被加载到主内存中。另外，原始消息的位置或 URL 保存在主内存数据结构中。如果需要，我们可以获得所要的消息。随着时间的推移，特征向量可能由于内容（见本章参考文献［MASU10a］）的动态特性而改变，因此特征集可能会演进。基于之前针对动态特征集中数据流演进的研究，我们还研究了替代方案。

26.3.3 RDF 仓库架构

RDF 是语义 Web 的数据形式。但是，它可以用来表示世界上任何链接数据。RDF 数据实际上是三元组集合。三元组包括 3 个部分：主体、谓词和客体。在 RDF 中，几乎所有的东西都是资源，因此格式其实就是名称。主体和谓词都是资源。客体可以是资源也可以是文字。在这里，RDF 数据可以被看作有向图，其中谓词是从主体到客体的边。因此，在我们的研究中对任何图形建模，都利用 RDF 三元组格式。在这里，在图形数据集中从源节点到目的地节点的边分别表示为 RDF 三元组主体、谓词和客体。为了降低 RDF 三元组的仓库大小，我们利用字典编码，即，把每个唯一字符串替换为唯一的编号并且以二进制格式存储 RDF 数据。因此，RDF 三元组将含有编码形式的主体、谓词和客体。我们维护一个单独的表/文件来跟踪字典编码信息。为了解决数据的动态性质，我们通过在网络中添加时间戳，以及表示信息的主体、谓词和客体来将 RDF 三

元组扩展为四元组。

图 26.3 显示了我们的仓库体系结构包含两个组件。图 26.3 的上半部分描述了数据预处理组件，下半部分展示的是应答查询组件。我们对数据的生成和预处理包含 3 个子部分。如果数据不是 N 个三元组，那么我们就使用 N 个，三元组转换组件把它转换成 N 个三元组序列化格式。PS 组件提取 N 个三元组数据，并把它们分割成谓词文件。基于谓词的文件被送到 POS 组件中，该组件将根据客体的类型，将谓词文件分割成更小的文件。

我们的 Map/Reduce 框架有 3 个子组件。它需要用户的 SPARQL 查询，并将其传递给输入选择器和计划生成器。该组件将选择输入文件，决定需要多少 Map/Reduce 作业，并将信息传递给使用 Map/Reduce 框架运行作业的连接执行程序组件。然后它会将来自 Hadoop 中的查询答案发给用户。

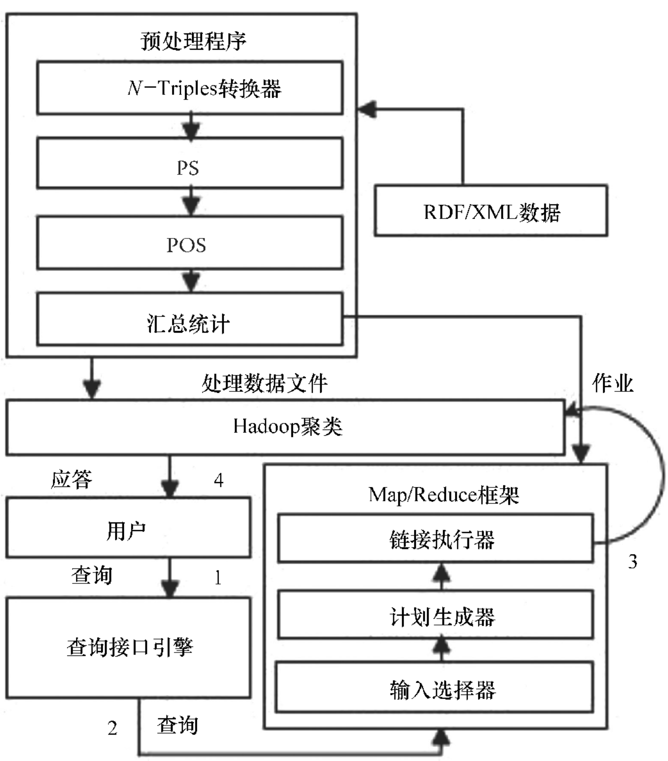


图 26.3　RDF 仓库架构

26.3.4　数据存储

我们以 N – Triples 格式存储数据，因为在这种格式中，我们在文件的一行中有一个

完整的 RDF 三元组（主体、谓词和客体），这与 Map/Reduce 作业一起使用会非常方便。我们对数据进行字典编码以提高效率。字词编码是指用唯一的二进制数替换文本字符串，这不仅减少存储所需的磁盘空间，而且因为处理原始数据类型比字符串匹配要快，这会导致查询应答速度快。下面描述获得预期格式数据的处理步骤。

26.3.4.1 文件组织

我们不把数据存储在单一文件中，因为在 Hadoop 和 Map/Reduce 框架中，一个文件是一个 Map/Reduce 作业的最小输入单元，在没有缓存的情况下，始终从磁盘中读取数据。如果我们把所有数据放在一个文件中，那么整个文件就被输入到每个查询的作业中去。相反，如果我们把数据分成多个小文件，那么这个分割就会用以下两个步骤进行。

26.3.4.2 谓词分割

在第一步骤中，我们根据谓词分割数据。在实际 RDF 数据集中，不同谓词的数量不会超过 100。此划分将立即使我们能够减少任何没有变量谓词的 SPARQL 查询的搜索空间。对于这样的查询，我们可以为每个谓词选择一个文件，并仅在这些文件上执行查询。为简单起见，我们用谓词命名文件，例如，包含谓词 *pl*：*pred* 的所有三元组放进名为 *pl* - *pred* 的文件夹。但是，如果一个三元组模式里含有变量谓词，并且我们不能确定客体的类型，就必须考虑所有文件。如果我们可以确定客体类型，那么我们就考虑那些含有这种客体类型的文件。

26.3.4.3 谓词客体分割

在接下来的一步中，我们要使用 rdf_ type 文件中显式类型的信息。该文件首先被分成与 rdf：type 谓词具有不同客体、但数量一样多的文件。客体值将不再需要存储在文件中，因为它们可以从文件名中轻松检索。这将进一步减少存储数据所需的空间。

然后我们根据客体的类型，划分剩下的谓词文件。并不是所有的客体都是 URI，有一些是文字。文字将会由谓词命名的文件保存，对于这些文字不需要做进一步处理。这些文件中的 URI 客体的信息类型并没有被提及，但是可以从 rdf - type_ ＊文件中检索。URI 对象将被移动到它们各自的以 predicate_ type 命名的文件中。

26.3.5 使用 Hadoop/Map/Reduce 技术应答查询

对于查询，我们可以使用类似于 SQL 的查询语言 Hive 和 RDF 数据的查询语言 SPARQL。当在 HiveQL 中提交查询时，在 Hadoop 上安装并运行 Hive，可以基于上面提供的架构来应答该查询。当提交 SPARQL 查询以从图形中检索相关数据时，首先我们生成一个具有最小数量 Hadoop 作业的查询计划。

接下来，我们运行作业并应答查询。最后，当我们向用户呈现查询结果时，要将用于编码字符串的数字转换回字符串。我们专注于最大限度地减少作业的数量，因为在我们的观察中，发现建立 Hadoop 任务是非常昂贵的，并且也是应答查询时的主要因素（按时间）。寻找最小数量作业的搜索空间是指数级的。所以，我们试图找到一个基于贪婪的解决方案，或者一般来说，寻找一个近似解决方案。我们的方法将能够处理涉及推理的查询。如果需要的话，我们就可以实现推断数据。

26.3.6　数据挖掘应用

为了检测异常/内部威胁，我们正在研究机器学习和域知识指导技术。我们的目标是建立一个相对的基准来评估混沌吸收子的有效性。我们采用基于半监督模型的整体方法，而不是将正常行为建模和检测变量作为异常模型。需要注意的是，首先，在我们机器学习的技术中，我们应用一系列活动或维度作为特征。其次，领域知识（例如敌对行为）将作为半监督学习的一部分，并且将用于识别正确的特征。最后，我们的技术将能够识别一个全新的异常。随着时间的推移，活动/维度可能改变或者偏离。因此，我们的分类模型需要适应，并确定新的类型或全新的异常。我们开发了自适应和新型的类检测技术，这使得我们的内部威胁检测可以应对变化，进而能识别或隔离新的异常。

我们采用分类技术来检测内部威胁/异常。每个不同的内部任务被视为类，维度/活动被视为特征。由于分类是有监督的任务，而我们需要一个训练集。给定一个训练集，特征提取将是一个挑战。我们应用 n – grams 分析来提取特征或产生一些基于时间属性的序列。一旦一个新的测试用例产生，我们首先测试它对我们的分类器模型的作用。对于分类器模型，我们可以应用支持向量机、KNN 和马尔科夫模型。

从机器学习的角度来看，习惯上把行为归类为异常或正常。然而，恶意内部人士的行为（即内部威胁）可能不会被立即识别为恶意行为，并且还可能也具有良性行为的微妙偏离。这种情况下，传统的基于机器学习的分类模型很可能将恶意内部威胁分类为良性的。能够看到基于机器学习的新类检测技术，可能检测到作为新类的内部威胁，并因此触发警报（见本章参考文献［MASU10a］），这将是一个非常有意义的事情。

新类检测技术被应用于从用户活动中生成的大量数据。由于这些数据具有时间特性，并且连续生成，所以通常称为数据流。新类检测模型随着输入数据逐步更新。这将允许我们将内存需求保持在一个一定的限制内，因为原始数据将被丢弃，但是行为的特征/模式仍将在模型中进行汇总。此外，这种增量学习也将减少训练时间，因为即使有新的输入数据，模型也不需要从头开始建立。因此，这种增量学习技术在实现可扩展性上是有用的。

我们正在研究我们已经开发的技术以及其他有关建模和异常检测的相关技术。特别是，我们正在开发：

■ 能够分析，并对良性和异常任务建模的工具。

■ 识别正确的维度和活动，并应用修剪以丢弃不相关维度的技术。

■ 用于应对变化和新类/异常检测的技术。

在一个典型的数据流分类任务中，往往假设类的总数是固定的。这种假设在内部威胁检测情况下可能是无效的，因为新的类可能演变。传统的数据流分类技术不能识别新的类的实例，直到手动识别新类的出现，并且该类的标记实例被呈现给用于训练的学习算法。由于概念漂移的存在，当底层数据分布随着时间变化时，问题就变得更具挑战性。我们已经开发出一种新的和高效的技术，这可以自动检测一个新类（即全新异常）的出现，这种检测需要通过量化未标记的测试实例之间的内聚，并且从训练实例中分离测试实例。我们的目标是运用可用的数据建立模型。

这个模式的一个有趣的方面是它可以捕捉任务维度的动态性质，并且滤除嘈杂的行为。维度（良性和异常）具有动态性质，因为它们倾向于随时间而变化，我们将其视为概念漂移。新类别检测的一个主要挑战是从概念漂移和噪声数据中分离出新颖的类。我们在目前的工作中正在探索这个挑战。

26.4 综合框架

正如我们在26.2节中论述的那样，内部威胁检测是一个非常具有挑战性的问题。在上一节中，我们讨论了处理这个问题的方法。内部威胁不仅发生在应用程序级，它还发生在所有级别，包括操作系统、数据库系统和应用程序。此外，由于内部程序不断变化的模式，使用一个纯静态算法检测所有类型的恶意行为是不可能的。所以必须需要一个动态学习方法。从本质上讲，我们需要一个全面的方法来解决内部威胁问题。但是，为了提供一个更全面的解决方案，我们就需要一个更全面的框架。因此，我们为内部威胁检测提出一个框架。我们的框架将实施一些相关的解决方案来检测内部恶意威胁。图26.4说明了这样一个框架。我们正在研究4个方法来解决这个问题。这个框架的核心模块是实现了嵌入引用监视器的技术，以方便实现特征收集。这个特征收集过程由两个模块组成；一种使用博弈论方法而另一种使用基于自然语言的方法，来确定哪些特征可以被收集。第四个模块使用机器学习技术分析收集的特征。总之，4种方法之间的关系可以表征如下：

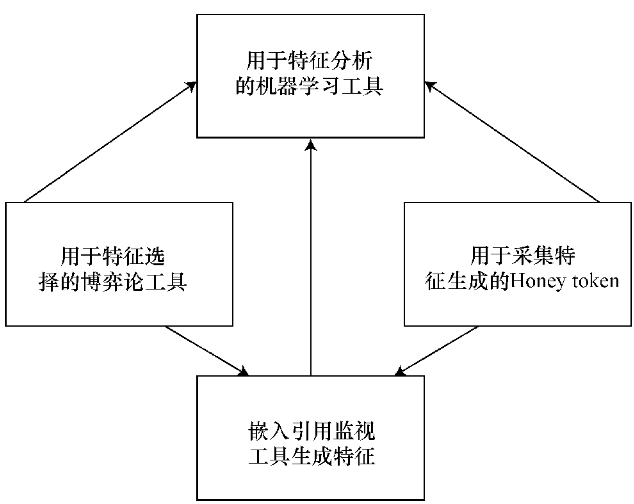


图26.4　内部威胁检测框架

■ 嵌入引用监视器（In-line reference monitors，IRM）进行隐蔽的，细粒度的特征收集。

■ 博弈论技术确定哪些特征需要被IRM收集。

■ 一般的自然语言处理技术和特殊的Honey token生成技术，将采取积极的方法来

引入可以收集的其他有用的功能（即 Honey token 访问）。

■ 机器学习技术将使用收集的特征来推断和分类恶意内部程序。

我们框架的细节由本章参考文献［HAML11］提供。我们在假设中嵌入引用监视工具、博弈论工具和 Honey token 生成工具，这些都将挑选和精炼我们需要的特征。随后，数据挖掘工具将分析特征，并且确定是否有潜在的内部威胁。

我们已经开始实施框架的部分工作。特别的，我们已经开发了一些数据流挖掘技术用来检测内部威胁。恶意内部活动的证据常常埋藏在大数据流中，例如累积数月或数年的系统日志。基于集成的数据流挖掘，并通过利用多分类器模型，可以在这些数据流中实现高精度的异常检测，即使数据流是无限的、不断变化的和未标记的。这使得该方法能有效地识别内部威胁，即使是那些通过随时间变化而改变他们行为来试图隐瞒自己活动的内部威胁。我们的方法是基于集成的数据流挖掘，无需监督学习。通过基于图形的异常检测的对比，来应对内部威胁检测的问题。证明基于集成的方法是显著的，且比传统单模型方法更有效。我们进一步探讨了不同学习策略对不断演变内部威胁的适用性。同时，也开发了可在云计算上实现的无监督机器学习算法，来检测内部威胁。更多的信息参见本章参考文献［PALL12］。基于集成的数据流挖掘应用信息参见第 25 章，算法的细节参见本章参考文献［MASU10a］。

26.5　总结和展望

在本章中，我们讨论了内部威胁检测方法。将内部人员和他们的沟通视为 RDF 图表，然后进行查询并且通过挖掘图形来提取信息。我们也提供了一个综合的框架来进行内部威胁检测。

内部威胁问题是一个刚刚开始被研究的，且具有挑战性的问题。其中的主要问题是内部威胁常常改变他/她的模式和行为，因此，我们需要自适应的工具。例如，我们在第 25 章讨论的数据流挖掘技术可被用于检测这种威胁。我们也已经有了一些初步的数据流挖掘工具（见本章参考文献［PALL12］）。因为大量的数据需要被分析，基于云计算的数据挖掘对内部威胁检测来说将是合适的方法。我们的方法对内部威胁检测基本上可以看作是安全即服务的解决方案。

参考文献

[AHME09] Ahmed, M.S. and L. Khan, SISC: A text classification approach using semi supervised subspace clustering. *DDDM '09: The 3rd International Workshop on Domain Driven Data Mining in Conjunction with ICDM 2009,* December 6, 2009, Miami, Florida.

[BERR07] Berry, M.W., M. Browne, A. Langville, V.P. Pauca, and R.J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1): 155–173, 2007.

[CARM09] Carminati, B., E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, A semantic web based framework for social network access control. *Proceedings of Symposium on Access Control Models and Technologies (SACMAT)*, Stresa, Italy, 177–186, 2009.

[COOK06] Cook, D. and L. Holder, *Mining Graph Data*, Wiley Interscience, New York, NY, 2006.

[EBER09] Eberle, W. and L. Holder, Applying graph-based anomaly detection approaches to the discovery of insider threats, *IEEE International Conference on Intelligence and Security Informatics (ISI)*, Dallas, TX, June 2009.

[HADO] http://hadoop.apache.org/

[HAML11] Hamlen, K., L. Khan, M. Kantarcioglu, V. Ng, and B. Thuraisingham, Insider threat detection, White paper, The University of Texas at Dallas, April 2011.

[MASU10a] Masud, M., J. Gao, L. Khan, J. Han, and B. Thuraisingham, Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering (TKDE), April 2010, IEEE Computer Society.*

[MASU10b] Masud, M., Q. Chen, J. Gao, L. Khan, J. Han, and B. Thuraisingham, Classification and novel class detection of data streams in a dynamic feature space, In: *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2010,* Barcelona, Spain, September 20–24, 2010, Springer 2010, ISBN 978-3-642-15882-7, 2: 337–352.

[MAYB05] Maybury, M., P. Chase, B. Cheikes, D. Brackney, S. Matzner, T. Hetherington, B. Wood et al., *Analysis and Detection of Malicious Insiders,* In: *2005 International Conference on Intelligence Analysis*, McLean, VA.

[MITC97] Mitchell, T., *Machine Learning*. McGraw-Hill, 1997.

[NEUM08] Neumann, T. and G. Weikum, RDF-3X: A RISC-style engine for RDF, *Proceedings of the VLDB Endowment,* 1(1): 647–659, 2008.

[PALL12] Parveen, P., N. McDaniel, V.S. Hariharan, B.M. Thuraisingham, and L. Khan, Unsupervised ensemble based learning for insider threat detection. *SocialCom/PASSAT Conference,* 718–727, Amsterdam, the Netherlands, 2012.

[THUR09] Thuraisingham, B., M. Kantarcioglu, and L. Khan, Building a geosocial semantic web for military stabilization and reconstruction operations. *PAISI*, Bangkok, Thailand, 1: 2009.

[TONG09] Tong, H., Fast algorithms for querying and mining large graphs. CMU Report, ML-09-112, September 2009.

第 27 章　以云为中心保障信息共享

27.1　概述

随着云计算的出现和软件即服务（Software as a Service，SaaS）模式的不断发展，这对保障信息共享（Assured Information Sharing，AIS）作为云中的服务提出了越来越多的需求。于2011 年4 月 NSA（国家安全局）的 CIO（首席信息官）Lonny Anderson 在描述该机构关注以“以云为中心”的方式与其他机构进行信息共享时，将这种需求的紧迫性已经表现出来（见本章参考文献［NSA11］）。同样地，国防部已经在利用云计算模式的高效经济、灵活和可扩展性，以满足其通过敏捷企业所提供信息的力量来确保任务成功的愿景，从而在整个信息环境中具有可操作性和自由度（见本章参考文献［DoD］、［DoD07］、［DoD09］）。因此，这两个机构认为云环境的有效 AIS 技术和工具都非常需要。

虽然一些 AIS 工具在过去五年中，作为基于策略的信息共享已经被开发（见本章参考文献［FINI09］、［THUR08］、［AWAD10］、［RAO08］），但据我们所知这些工具都没有运行在云上，并且也不提供可扩展性，去支持大量用户使用大量数据。我们最近用于支持基于云的 AIS 的原型系统，已经应用了以云为中心的引擎，通过非云策略引擎以 XACML 格式执行策略，能够查询关系数据库中的大量数据（见本章参考文献［THUR10］、［THUR11］）。虽然相对于之前的努力来说，这是一个显著的改善（并给予我们机会实施云计算的方案），但它始终至少有三个明显的局限性。第一，基于 XACML 的策略规范，不足以支持像 NSA 和国防部这样的 AIS 任务所需的许多复杂策略。第二，为了满足关键任务的可扩展性和效率要求，策略引擎需要在云中运行。第三，基于关系技术的查询处理安全存在局限性，主要表现在表示和处理许多应用所需的非结构化数据方面。

为了安全有效地共享大量数据，显然在云中需要无缝集成策略和数据管理器。因此，为了满足云中心 AIS 的需求，我们需要①云驻留策略管理器，强制执行语义丰富的信息共享策略；②云驻留数据管理器，它能安全地存储和检索数据，并与策略管理器无缝集成。据我们所知，目前还没有这样的系统存在。因此，我们基于云计算 AIS 系统的项目设计和开发分为两个阶段进行。

我们已经设计了一个系统，并且实现了以云为中心的信息共享保障系统（Cloud－centric Assured Information Sharing System，CAISS）的一个版本，主要利用我们之前设计的内部工具和一些开源工具作为系统组件。CAISS 包括两部分：一个是以云为中心的策略管理器并执行 RDF 中的策略，另一个是以云为中心的数据管理器将存储和管理 RDF

中指定的数据。RDF 数据管理器本质上是一个 SPARQL 查询引擎，一种被 Web 网络技术交流社区用来查询 RDF 数据广泛使用的语言。RDF 是语义 Web 语言，在制定与推理策略方面，一般认为与 XACML 相比其更具表现力。此外，我们的策略管理器和数据管理器将无缝集成，因为它们都管理 RDF 数据。在第一阶段，我们为云中心的 AIS 选择基于 RDF 的方法，因为它满足了前面提到的两个必要条件，已经开发了一个基于 RDF 的非以云为中心的策略管理器（见本章参考文献［CADE11a］）和一个基于 RDF 的云中心数据管理器（见本章参考文献［HUSA11］）。具体来说，我们正在加强基于 RDF 的策略引擎，在云端运行，扩展我们以云为中心的 RDF 数据管理器与策略管理器集成，并为 CAISS 构建一个集成框架。我们在第 28 章描述 CAISS 版本的详细实现。

虽然我们的 CAISS 设计和实施将会是第一个支持以云为中心的 AIS 的系统，但它仅在单个可信赖云上运行，因此不支持跨多个云的信息共享。此外，虽然 CAISS 采用的基于 RDF 的策略规范和形式语义学方法，此方法将比基于 XACML 的方法明显更具表现力，但是由于 RDF 不提供足够丰富的词汇表（例如，对类的支持），它就不会支持有关内容的增强的机器可解释性。我们因此设计了一个功能齐全和功能强大的 AIS 系统（CAISS + +）来解决这些不足之处。CAISS 对 CAISS + + 来说是重要的基础，因为 CAISS 可以被用来作为基准框架，并可以作为几个性能尺度的参照物，比如存储模型的效率和基于 OWL 策略的表现力。此外，由于 CAISS 和 CAISS + + 共享相同的核心组件（策略引擎和查询处理器），CAISS 中这些组件的实现和集成的经验，对于开发 CAISS + + 是宝贵的。最终，CAISS 的测试和评估，也为我们提供了对 CAISS 不足之处的重要见解，这将在 CAISS + + 的实施中系统地解决。

我们还将对策略规范和软件级保护机制进行形式分析，强制执行这些机制，为所得到的系统提供极高的安全保障。我们设想 CAISS + + 应用在高度关键任务的应用程序。因此，必须提供保证策略是以可靠的方式执行的。在对 CAISS + + 分析中，我们将利用形式策略分析（见本章参考文献［JONE10］、［JONE11］）中的相关工作，并通过机器认证的嵌入引用监视器（见本章参考文献［HAML06a］、［HAML06b］、［SRID10］）去实施。此类分析将用于建模和验证安全属性，主要是 CAISS + + 的可信计算基中的核心软件组件实施时的安全属性。

CAISS + + 对信息共享来说将会是一项关键技术，因为它是一种组合，不仅采用以云为中心的策略规范，并实施以云为中心的数据存储和有效的查询评估。CAISS + + 将利用本体语言 OWL 来构建策略。这种本体与基于语义的网络规则语言（例如，SWRL）的混合，促进了对策略的分散推理来实现其安全性。此外，CAISS + + 将包括一个基于信息共享策略的 RDF 处理引擎，它提供基于成本优化来评估 SPARQL 查询。

本章结构如下。我们将在 27.2.1 节讨论 CAISS 的设计，27.2.2 节讨论 CAISS + + 的设计。27.2.3 节和 27.2.4 节分别讨论形式策略分析和 CAISS + + 实施方法。相关工作在 27.3 节进行讨论。在 27.4 节对本章进行总结。图 27.1 说明了本章包含的内容。在本章参考文献［THUR12］中有我们工作的细节。我们的 AIS 方法由附录 D 提供。需要注意的是，第 28 章将讨论 CAISS 作为语义 Web 服务集合的实现版本。

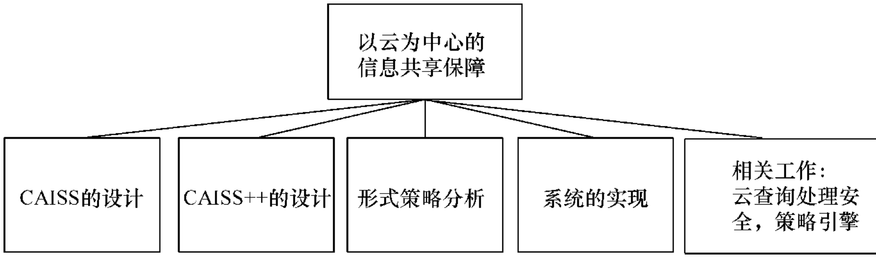


图 27.1　以云为中心的信息共享保障（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar, V., Rachapalli, J., Cadenhead, T., Kantarcioglu, M., Hamlen, K. W., Khan, L., 和 Husain, M. F. Media from Lecture Notes in Computer Science, Proceedings of Intelligence and Security Informatics - Pacific Asia Workshop, PAISI 2012, Kuala Lumpur, Malaysia, 7299, 2012, p. 1 - 26, © Springer, ISBN 978-3-642-30427-9.）

27.2　系统设计

27.2.1　CAISS 设计

我们正在以下两个方面加强我们的工具。①使用语义 Web 数据来保护云查询处理；②基于语义 Web 的策略引擎来开发 CAISS。我们的工具详情见 27.4 节（相关工作）。在本节中，我们将讨论对我们开发 CAISS 所用工具的改进。

首先，我们基于 RDF 的策略引擎对表示为 RDF 图的数据执行访问控制、编辑和推理控制策略。其次，我们的云 SPARQL RDF 数据查询引擎使用 Hadoop / Map/Reduce 框架。需要注意的是，Hadoop 是 Apache 分布式系统，Map/Reduce 位于 Hadoop 之上，并执行作业调度。如我们基于云的关系查询处理器原型（见本章参考文献［THUR10］）的情况，SPARQL 查询引擎还处理 XACML 中指定的策略，策略引擎实现 XACML 协议。使用 XACML 作为策略语言，这需要对 XACML 设计中所使用的一般概念要有广泛的了解。因此，XACML 中的策略制定需要一个陡峭的学习曲线，因此应该由一个经验丰富的管理员负责该任务。使用 XACML 的第二个缺点是和性能相关。XACML 的当前实现需要针对系统中的每个策略来评估访问请求，直到策略应用于输入的请求。对于相对较少的用户和少量访问请求的系统，这种策略是足够的。然而，对于具有大量用户和大量访问请求的系统，上述策略成为性能瓶颈。最后，XACML 对于捕获信息共享策略的语义不是很有表达能力。此前的工作表明，基于 Web 的语义策略表达能力更强。这是因为语义 Web 技术是基于描述逻辑的，并且有能力来代表知识以及知识的推理。因此，我们的第一步是用一种基于 Web 的语义策略引擎替换基于 XACML 的策略引擎。因为我们已经有基于 RDF 的策略引擎，对于第 1 阶段的原型，我们将增强引擎并将其与 SPARQL 查询处理器集成。由于我们的策略引擎是基于 RDF 的，并且我们的查询处理器也管理大型 RDF 图，因此数据和策略之间不会有阻碍不匹配的情况。

27.2.1.1 增强策略引擎

我们目前的策略引擎有一个局限性，就是它不运行在云上。因此，我们将 RDF 策略引擎移植到云环境中，并将其与 SPARQL 查询引擎集成为云中的联合查询处理。我们的策略引擎将从 Hadoop/Map/Reduce 框架的可扩展性和分布式平台中受益，以通过大型分布式 RDF 三元组存储（数十亿的 RDF 三元组）来响应 SPARQL 查询。使用 RDF 作为我们的数据模型，理由是：①RDF 使我们能够实现每个机构/组织分别编目的看似不同的信息源之间的数据互操作性；②使用 RDF 允许参与机构创建以数据为中心的应用程序，这些应用程序可利用现有的集成数据；③由于 RDF 不需要使用明确的数据生成模式，因此可以轻松适应不断变化的用户需求。策略引擎的灵活性是基于其接受的高级策略，并通过数据的有向 RDF 图表示形式把它们作为查询规则执行。虽然我们以前的工作侧重于来源数据和访问控制策略，但我们的 CAISS 原型将足够灵活地处理 RDF 中的数据，并将包括信息共享策略。我们的策略引擎的优势在于，它可以处理由 RDF 表示的任何类型的策略和逻辑规则。

我们策略引擎的第二个局限，是它目前只能解决一定类型的策略，比如机密性、隐私和修订策略。我们需要把信息共享纳入策略引擎。然而，我们对基于激励的 AIS 以及在云中 AIS 原型也进行了仿真研究。其中，已经定义了一系列信息共享策略，比如“如果英国不与印度共享信息，美国则提供信息给英国”。我们在 RDF 中指定这些策略并将它们纳入增强的策略引擎中处理。

27.2.1.2 增强 SPARQL 查询处理器

虽然我们有一个工具将通过 Hadoop 上的大型 RDF 图来执行 SPARQL 查询（在第 13 章中讨论），但仍然需要支持路径查询（即，对 RDF 图中的路径请求提供答案的 SPARQL 查询）。一个 RDF 三元组可以被视为一个从主体到客体的弧，这个弧是用谓词标记的。SPARQL 查询的答案是基于图中可达性的（即源节点和目标节点之间的路径）。沿着一条路径的弧上的标签级联，可以被认为是属于路径查询中答案集的单词。单词的每个部分均由 RDF 图中的三元组的一些谓词标签给出。我们设计了一种算法来确定候选三元组作为分布式 RDF 图中的答案集。首先，将 RDF 文档转换为基于谓词标签进行拆分的 N 个三元组的文件。一个词中的术语可以对应于一些谓词文件。其次，我们通过在分布式 RDF 图中追踪适当的路径形成这个词。我们使用 Map/Reduce 作业构建单词，并将候选 RDF 三元组作为一个指令集。最后，返回所有一组有序的 RDF 三元组作为对应的 SPARQL 查询的答案。

27.2.1.3 集成框架

图 27.2 提供了 CAISS 体系结构的概述。以云为中心的 RDF 策略引擎与增强型 SPARQL 查询处理器的集成必须解决以下问题。首先，我们需要确保基于 RDF 策略可以存储在查询处理器已使用的现有存储模式中。其次，我们需要确保增强型查询处理器能够高效地评估基础 RDF 存储器上的策略（即路径查询）。最后，我们需要对 CAISS 进行绩效评估，以验证其符合各参与机构的绩效要求。图 27.3 说明了 CAISS 操作的概念。在这里，多个机构将在单个云中共享数据。增强的策略引擎和以云为中心的 SPARQL 查询处理器将执行信息共享策略。

在开发 CAISS ++之前，完成比如 CAISS 等概念原型的证明有几个好处。首先，CAISS 本身可以在单个云中共享数据。其次，在实施 CAISS ++时，我们将有一个基准

系统，可以在效率和易用性方面进行比较。第三，这将为我们提供宝贵的经验教训，将不同部分的 AIS 融入云中。最后，通过在 CAISS 上运行不同的方案，可以确定 CAISS ＋＋中需要解决的潜在性能瓶颈。

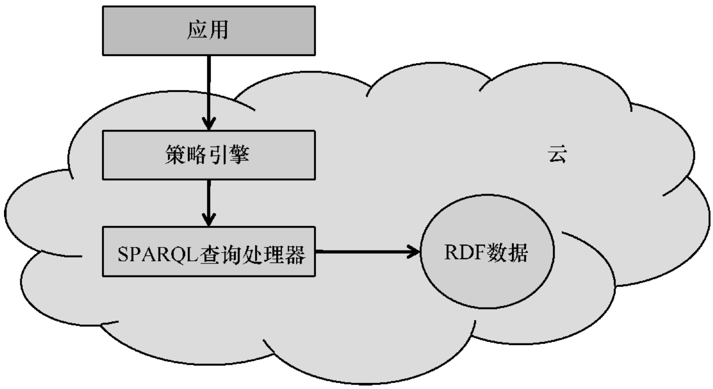


图 27.2　CAISS 原型概述（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L.，and Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 –26，© Springer，ISBN 978 –3 –642 –30427 –9. ）

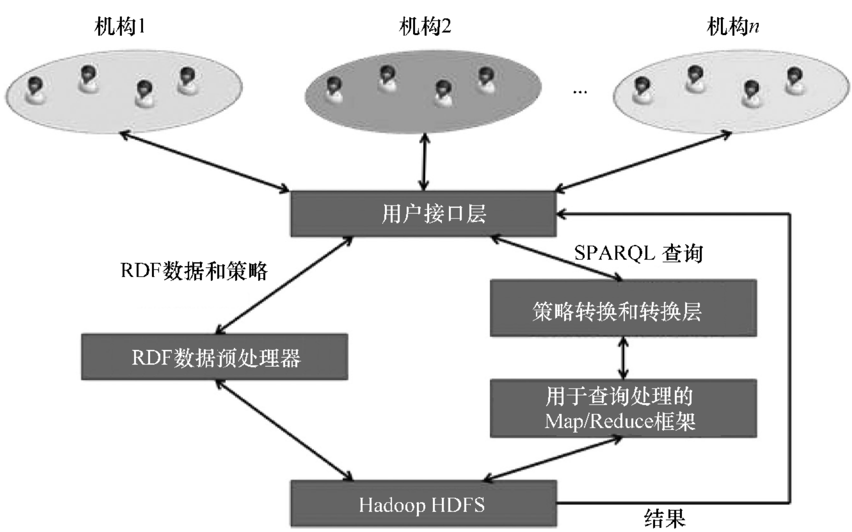


图 27.3　CAISS 操作（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L.，and Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 –26，© Springer，ISBN 978 –3 –642 –30427 –9. ）

27.2.2 CAISS + +的设计

我们已经审查了候选方案，并进行了 CAISS + +的初步设计。根据从 CAISS 原型中吸取的经验和 CAISS + +的初步设计，我们将对 CAISS + +进行详细的设计，并在第 2 阶段中实施 CAISS + +的操作原型。在本节中，我们将讨论一些 CAISS 的局限性，然后讨论 CAISS + +的设计方案。

27.2.2.1 CAISS 的局限性

1）策略引擎：CAISS 使用一个基于 RDF 的策略引擎，但其表达能力有限。RDF 的目的是提供一个描述资源的结构（或框架）。OWL 建立在 RDF 之上，并且它被设计用于需要处理信息内容的应用程序，而不仅仅是向人类用户提供信息。OWL 通过提供用于描述属性、类以及形式语义的附加词汇，增加了比 RDF 环境下对内容的机器理解性。OWL 有 3 种越来越有表现力的语言：OWL Lite、OWL DL 和 OWL Full，可以根据应用需求自由选择合适的子语言。在 CAISS + +中，我们计划利用比 RDF 更有表现力的 OWL，通过组织特定的域本体和系统范围的上层本体，来建立安全策略（需要注意，CAISS + +将重用组织的现有域本体，或不存在本体时，促进创建一个新的领域本体。还有就是，我们必须设计上层本体，它将被 CAISS + +集中的组件使用）。此外，CAISS + +将利用分布式推理算法，利用本体来实施安全策略。

2）Hadoop 存储架构：CAISS 使用静态存储模型，其中用户只在初始化阶段把 RDF 数据提供给系统。此后，用户不允许更新现有数据。在 CAISS + +中，允许用户对存储在 HDFS 中的现有 RDF 数据添加新数据。需要注意的是，仅允许用户附加新数据，而不是删除/修改现有数据。当然这些数据来自 HDFS 执行的、具备只能添加限制的文件。

3）SPARQL 查询处理器：CAISS 仅仅支持简单的 SPARQL 查询，也就是那些利用基本图形模式（BGP）的查询。但是，在 CAISS + +中支持其他 SPARQL 查询操作，比如 FILTER、GROUP BY、ORDER BY 等。另外，CAISS 使用启发式查询优化器，旨在减少回答一个查询时所需要的 Map/Reduce 作业数目。CAISS + +将引入一个基于成本的查询优化器，最大限度地减少查询执行过程中访问的三元组数量。

27.2.2.2 CAISS + +设计

CAISS + +克服了 CAISS 的局限性。对 CAISS + +的设计和实施细节将在阶段 2 中执行。从 CAISS 中学到的经验，将会促进 CAISS + +的详细设计。我们假设数据用适当的 DoD 加密技术加密，并因此对本项目不进行加密研究。对 CAISS + +操作的概念与其他参与机构在图 27.4 中展示，多个组织在单一云中共享数据。

CAISS + +的设计是一种基于 OWL 策略引擎和 RDF 处理引擎的新型组合。因此，这项设计由几个任务组成，其中每个任务单独完成后，所有的任务将被集成到一个单一的框架：①基于 OWL 的策略引擎，策略引擎使用一组特定域的本体和上层本体构建 AIS 任务的策略。任务执行的策略可能需要一个分布式推理，因此，我们将评估现有的分布式推理器。②RDF 处理引擎，处理引擎需要构建复杂的存储架构以及高效的查询处理器。③集成框架：最终的任务是将策略引擎和处理引擎结合成一个集成框架。

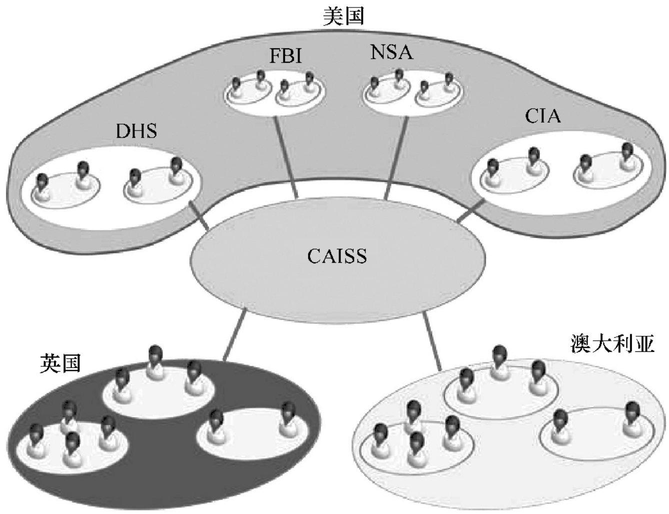


图 27.4　CAISS + + 场景（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L.，和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 – 26，© Springer，ISBN 978 – 3 – 642 – 30427 – 9）

CAISS + + 的初始设计在设计的简单性与可扩展性和效率之间进行了权衡。第一个候选设计方案被称为集中 CAISS + +，简单性是权衡的结果。而第二个候选设计方案（称为分散 CAISS + +）选择可扩展性和效率作为权衡的结果。最终，我们还提供一个混合 CAISS + + 架构试图结合这两者的优点。由于 CAISS + + 遵循需求驱动设计，我们通过上面提到任务的划分，来实现 AIS，这将在接下来的方法中有所展现。

27. 2. 2. 3　集中式 CAISS + +

图 27.5 说明了通过集中式 CAISS + + 交互的两个机构。集中式 CAISS + + 由共享云存储组成，用于存储共享数据。所有参与机构都存储各自的知识基础，这些知识基础由带有相应实例数据的本体组成。集中 CAISS + + 还包括一个上层本体、一个查询引擎（Query Engine，QE）和一个分布式推理器（Distributed Reasoner，DR）。上层本体用于捕获通常跨越参与机构所在领域的领域知识，而领域本体捕获给定的机构或领域的知识。需要注意的是，给定机构的领域本体将受到其他参与机构的域本体的保护。策略可以在上层本体或任何领域本体中被捕获，具体取决于它们的适用范围。值得注意的是，给定机构的领域本体将受到其他参与机构的领域本体的保护。

上层本体的设计以及捕捉参与机构要求的领域本体是一个重要的研究领域，是本体论工程问题的重点。这里，将使用基于描述逻辑中合适的 OWL 本地语言创建本体。描述逻辑通常是一阶逻辑的可判定部分，并将其作为提供声音形式语义的基础。推理在本体论中代表知识，并通过使用现有的优化推理算法进行。查询应答将利用推理算法来制

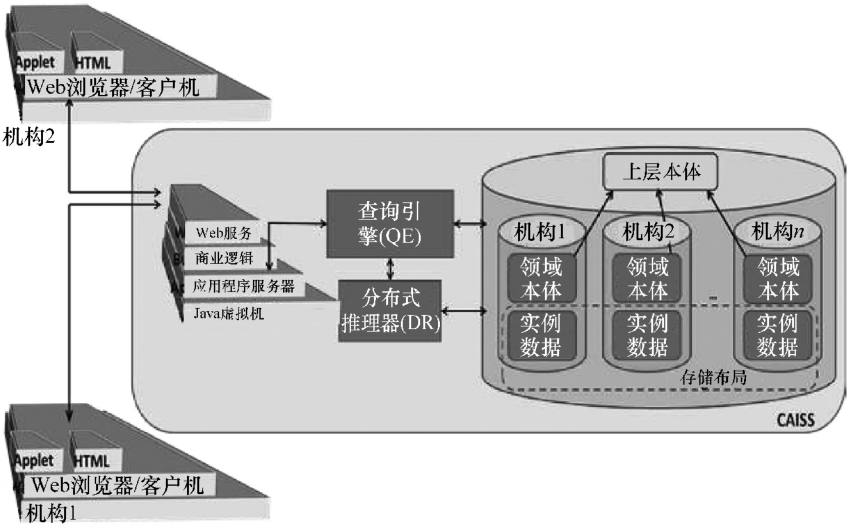


图 27.5 集中式 CAISS + +（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 – 26，© Springer，ISBN 978 – 3 – 642 – 30427 – 9）

定和回答智能查询。另外，OWL 中的策略编码将确保以可靠的方式强制执行。接下来，我们提出在得克萨斯大学达拉斯分校正在开展的一个研究项目，项目重点是提供一个总体框架，以一种已经被证明为正确的方式且使用相同的底层技术执行策略。这项工作可以用于建模和执行 CAISS + + 中的安全策略。项目中的实例数据，可以选择几种可用的数据存储格式（稍后讨论）。QE 从参与机构接收查询、解析查询，并确定计算是否需要使用 DR。如果查询简单，并且不需要使用推理器，则查询引擎将通过共享知识库直接执行查询。查询结果一旦出来，就将返回给查询机构。然而，如果查询是复杂的，并且需要对给定数据进行推理，则查询引擎就使用分布式推理器来计算推论，然后将结果返回给查询机构。分布式 DL 推理器与传统的 DL 推理器，在使用 Map/Reduce 框架执行云数据存储的推理能力方面有所不同。在第一阶段 CAISS + + 的初步设计过程中，我们将使用 LUBM 等现有基准，来对可用的分布式推理器进行全面的调查研究（见本章参考文献［GUO05］）。这个调查的目标是确定我们是否可以使用现有的任意推理器，或者是否需要构建我们自己的分布式推理器。在图 27.4 中，机构被表示为由 Web 浏览器、小应用程序和 HTML 组成的堆栈。一个机构可以使用 Web 浏览器将查询发送给 CAISS + +，它是由查询处理器处理的。

集中式 CAISS + + 与 CAISS 之间的主要区别在于：①CAISS 将使用 RDF 对安全策略进行编码，而集中式 CAISS + + 将使用比 RDF 更具表达性的、更合适的 OWL 子语

言，从而更好地捕获安全策略；②CAISS 中的 SPARQL 查询处理器将支持 SPARQL 表达性的有限子集，即仅提供对基本图形模式（Basic Graph Patterns，BGP）的支持，而集中式 CAISS ＋＋中的 SPARQL 查询处理器将被设计为支持最大表达能力的 SPARQL；③CAISS中使用的 Hadoop 存储架构仅支持在初始化步骤期间的数据插入。但是，当需要更新数据时，会删除整个 RDF 图，并在其中插入一个新的数据集。另一方面，集中式 CAISS ＋＋除了支持以前的功能外，还为用户打开了 Hadoop HDFS 的附加功能。此功能允许用户将新信息附加到以前上传到系统的数据中。

27.2.2.4 分散式 CAISS＋＋

图 27.6 说明了在分散式 CAISS＋＋中相互作用的两个机构。分散式 CAISS＋＋包括两个部分：即全局 CAISS＋＋和本地 CAISS＋＋。全局 CAISS ＋＋由共享云存储组成，参与机构仅使用其存储的各自的领域本体而不是集中式 CAISS ＋＋的实例数据。这和集中 CAISS＋＋不同。需注意到领域本体对不同组织可能是敏感的，因此，CAISS＋＋将使用自身领域本体来保护参与机构不去访问其他领域本体。当一个机构的用户查询 CAISS＋＋数据存储，全局 CAISS＋＋使用两个步骤处理查询。第一步，它执行检查，来验证用户是否被授权进行查询中的指定行为。如果第一步结果验证用户为授权用户，然后继续进行查询处理的第二步。在第二步中，全局 CAISS＋＋分配真正地参与机构进行查询，查询由参与机构所在的本地 CAISS＋＋处理。计算结果随后返回到全局 CAISS ＋＋，并将最终汇总结果返回给用户。如果数据需要跨域查询，查询处理的第二步就可能涉及查询分割。在这种情况下，来自不同机构（他们的本地 CAISS＋＋）的子查询结果需要组合进行查询处理。一旦合并结果，并计算最终结果，再将结果返回给查询机构

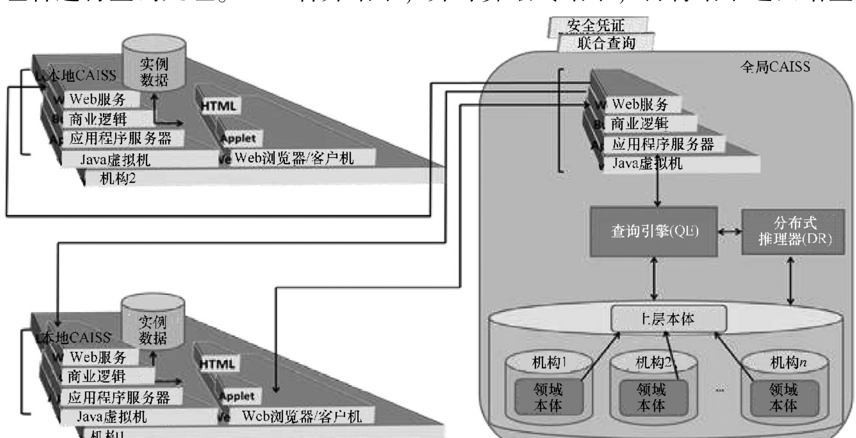


图 27.6 分散式 CAISS＋＋（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics－Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1－26，© Springer，ISBN 978－3－642－30427－9）

的用户。该图显示出了具有两个堆栈的机构，其中一个对应于本地 CAISS ++，另一个对应于 Web 浏览器、小应用程序和 HTML，由机构用于查询全局 CAISS ++。表 27.1 显示了集中式 CAISS ++方法的优缺点，而表 27.2 显示了分散式 CAISS ++方法的优缺点。

表 27.1　集中式 CAISS ++优缺点

优点	缺点
方法简单	难以更新数据，方法昂贵，因为数据需要在每次更新或一组更新时迁移到中央存储器
实施简单	导致数据重复
查询简单	如果数据是可用的不同格式，需要转化成 RDF 格式

表 27.2　分散式 CAISS ++的优缺点

优点	缺点
没有重复数据	查询过程复杂
扩展性和灵活性好	实施困难
高效	可能需要查询重写和查询分割

27.2.2.5　混合 CAISS ++

图 27.7 给出了混合 CAISS ++的概览，它结合了集中式 CAISS ++以及分散式 CAISS ++的优势。混合式 CAISS ++架构如图 27.8 所示。这是一个灵活的设计方案，因为参与机构的用户可以自由选择集中式 CAISS ++或分散式 CAISS ++。混合 CAISS ++由全局 CAISS ++和位于每个参与机构的一组本地 CAISS ++组成。全局 CAISS ++由共享云存储组成，由参与机构用于存储他们想与其他机构共享的数据。

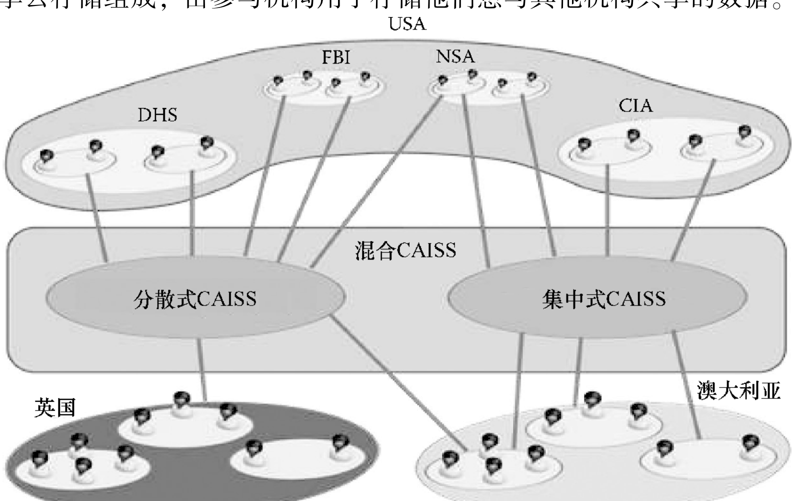


图 27.7　混合 CAISS ++概览（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L.，和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 – 26，© Springer，ISBN 978 – 3 – 642 – 30427 – 9）

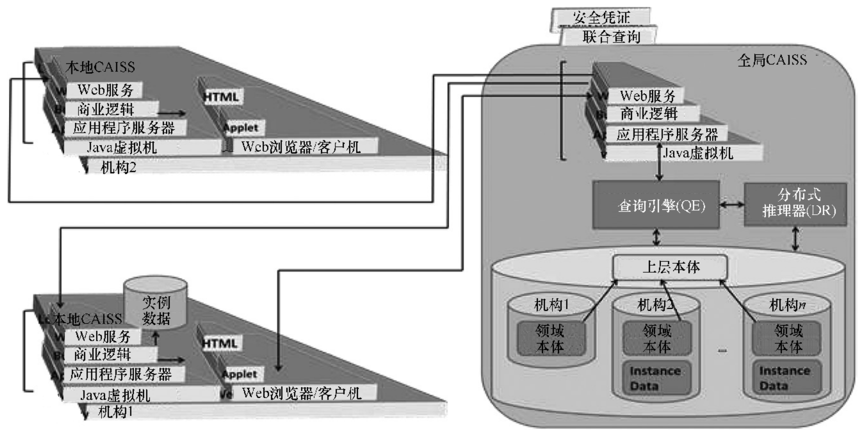


图 27.8 混合 CAISS + +架构（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L.，和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 – 26，© Springer，ISBN 978 – 3 – 642 – 30427 – 9）

一个机构的本地 CAISS + +用来接收和处理位于本地机构实例信息的联合查询。一个参与组由来自几个机构的用户组成，他们希望彼此能分享信息。小组成员决定采取集中或分散的方法。如果需要，另外的用户可以在稍后的时间点加入小组。混合 CAISS + +将被设计同时支持一系列参与组。此外，一个用户可以同时属于多个参与组。我们描述一些用例场景来说明这些操作。

1）第一个案例对应的场景，一组用户想要安全地分享彼此的信息而选择一个集中式方法。假设机构 1 中的用户想要和机构 2 中的用户共享信息，反之亦然，然后双方机构在全局 CAISS + +共享云上存储他们的知识库，包括领域本体和实例数据。集中式 CAISS + +方法使参与机构间相互信任地使用中心云作为存储。随后，信息共享在集中式 CAISS + +中进行。

2）第二种对应情况是一组用户选择分散式方法。例如，机构 3、4、5 想要互相共享数据并且相互选择分散式方法。所有 3 个机构在中心云存储上存储各自的域本体，并且这些信息只能被小组成员访问。随后的信息共享过程按照分散式 CAISS + +方法的方式进行。

3）第三种对应于一个机构的用户属于多个参与组的场景，其中一些选择集中式方法，而另一些则选择分散式方法。由于一部分用户是使用集中式方法进行共享的组，所以他/她需要通过将他/她的数据发送到中央云存储来使他/她的数据可用于该组。此外，由于另一部分用户是使用分散式方法进行共享的组，所以他/她需要在他/她所在的机构的本地 CAISS + +的帮助下回应联合查询。

表 27.3 显示了不同方法之间的权衡，这将使用户根据他们应用的需要，来选择合适的 AIS 方法。接下来，我们将描述云存储机制的细节，主要利用 Hadoop 来存储各机构的知识基础。然后，讨论云存储上分布式 SPARQL 查询处理的细节。

表 27.3　3 种基于功能 Hadoop 存储体系结构的比较

功能	集中式 CAISS + +	分散式 CAISS + +	混合 CAISS + +
无数据重复	×	√	可能
灵活性	×	×	√
扩展性	×	√	√
高效性	√	√	√
简洁性 - 无查询重写	√	×	×
可信的中心云数据存储	√	×	×

如图 27.9 所示，我们介绍了基于 Hadoop 的 RDF 存储和检索框架的架构概况。我们使用“存储”的概念为存储在基础 HDFS 中的 RDF 图，提供数据加载和查询功能。一个存储代表一个 RDF 数据集，因此可以包含多个 RDF 图，每个 RDF 图都有自己独立的布局。然后将 RDF 图上的所有操作隐式地转换为底层布局的操作，包括：

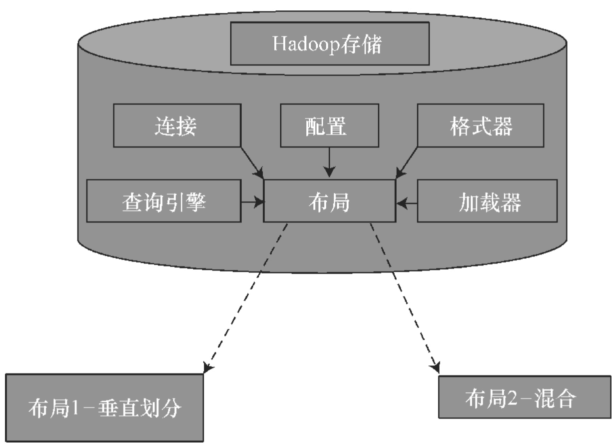


图 27.9　CAISS + + 使用的 Hadoop 存储体系结构（得到了 Springer 这一科学和商业媒体的允许，内容来自 Science + Busimess Media：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L.，和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics - Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 - 26，© Springer，ISBN 978 - 3 - 642 - 30427 - 9）

■ 布局格式器：执行格式化布局的功能，它是删除 RDF 图中所有三元组的过程，同时又保留用于存储图形的目录结构。

■ 加载器：将三元组加载到布局。

■ 查询引擎：允许用户使用 SPARQL 查询一个布局。由于我们的框架运行在底层 HDFS，所以在布局上的查询机制，包括将一个 SPARQL 查询转换成 Map/Reduce 作业的一个可能的管道，并在布局上执行这个管道。

■ 连接：维护和底层 HDFS 的必要的连接和配置。

■ 配置：维护配置信息，比如构成一个存储的每个 RDF 图的名字。

由于 RDF 数据作为我们存储架构的一部分，将存储在不同的文件夹中，因此对这

些文件夹和文件，需要采用某些命名规则。

27.2.2.6　命名规则

在我们的框架中，一个 Hadoop 存储可以由几个不同的 RDF 图组成。因此我们在 HDFS 中会为每一个 Hadoop 存储创建一个单独的文件夹。此文件夹的名称将对应于给定存储选择的名称。此外，根据在我们的框架中被分割成几个文件所选择的存储来布局 RDF 图。因此，一个单独的文件夹在 HDFS 中对每个不同的 RDF 图进行创建。该文件夹的名称被定义为默认 RDF 图的“默认值”，但是对于一个已命名的 RDF 图，图的 URI 将作为文件夹的名称。在我们的框架中使用抽象的存储，原因是这将简化属于各机构的数据管理问题。我们的框架支持两种布局，这些布局可以使用不同数量的 HDFS 文件来存储 RDF 数据

27.2.2.7　垂直分区布局

图 27.10 显示了垂直分区布局的存储模式。对于 RDF 图中包含的每个唯一的谓词，此布局将在基础 HDFS 中，使用谓词名称作为文件名来创建一个单独的文件。需要注意的是，只有名称中使用谓词 URI（通用资源标识符）的本地名称部分，并且文件名和谓词 URI 之间存在单独的映射。对于一个给定谓词的文件，存在一行对应每个包含该谓词的三元组。该行存储构成三元组的主体和客体。此模式将导致存储空间会明显地减少，因为谓词名称移动到文件名称，这将完全消除该谓词值的存储。但是，URI 或文字值的多次出现，将被多次存储在所有文件以及一个文件中。另外，SPARQL 查询可能需要查找多个文件，来确保一个完整的结果返回给用户。例如，一个查询需要查找属于一个特定主体或客体的所有三元组。

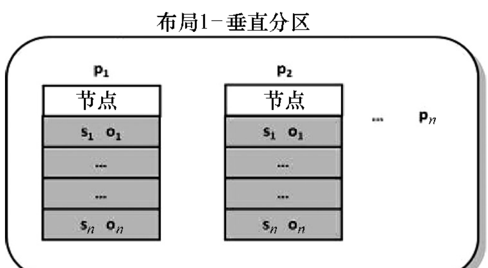


图 27.10　垂直分区布局（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics－Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1－26，© Springer，ISBN 978－3－642－30427－9）

27.2.2.8　混合布局

图 27.11 给出了混合布局的存储模式，此布局是垂直分区布局的扩展，由于除了为 RDF 图中每个独特的谓词创建单独的文件，它也创建了一个 SPO（主体、谓词、客体）格式的包含所有三元组的单独三元组文件。此文件的一个优点是它可以直接给出属于某

一特定主体或客体的所有三元组。这样的搜索操作需要在垂直分区布局中扫描所有文件。这种布局的存储空间效率不如垂直分区布局好，这是因为有附加的三元组文件。但是，对于一个 SPARQL 查询，要发现属于特定主体或客体的所有三元组来说，这种布局是更有效的。

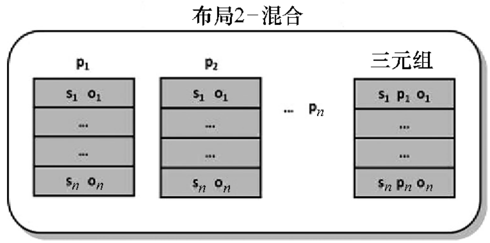


图 27. 11　混合布局（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics－Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1－26，© Springer，ISBN 978－3－642－30427－9）

27. 2. 2. 9　SPARQL 的分布式处理

CAISS ＋＋中的查询处理包含几个步骤（见图 27. 12）。第一步是查询解析和翻译，其中首先解析给定的 SPARQL 查询，以验证语法正确性，然后构建与输入查询相对应的解析树。解析树再转化成 SPARQL 代数表达式。由于给定的 SPARQL 查询可以具有多个等效的 SPARQL 代数表达式，所以，我们使用如何评估表达式中每个操作的说明，来注释每个这样的表达式。这种注释的 SPARQL 代数表达式与作为优化器输入的查询评估计划相对应。优化器选择最小化查询评估成本的查询计划。要优化查询，优化器就必须知道每个操作的成本。要计算每个操作的成本，优化器使用存储与 RDF 数据相关联的统计信息的元存储。根据 Map/Reduce 作业的数量，或作为查询执行的一部分进行访问的三元组的数量，可以测定给出的查询评估计划的开销。一旦选择了查询计划，将使用该计划评估查询，并输出查询结果。由于我们使用以云为中心的框架来存储 RDF 数据，所以评估引擎需要将 SPARQL 代数运算符转换为底层存储布局上的相应 Map/Reduce 作业。因此，在 CAISS ＋＋中，我们将为每个 SPARQL 代数运算符实现 Map/Reduce 作业。另外，评估引擎使用分布式推理器来计算查询评估所需的推论。

27. 2. 2. 10　框架集成

我们概述的组件是 CAISS ＋＋中的一部分，这些组件需要与其他组件结合使用。此外，这种集成过程取决于在 CAISS ＋＋提供的 3 种可能的设计方案中，用户所进行的选择，即集中式 CAISS ＋＋、分散式 CAISS ＋＋或混合 CAISS ＋＋。目前已经提交的各种 CAISS ＋＋的集成需要考虑到几个问题。首先，我们需要确保我们的本体工程实施的过程中，在捕获各个组织的需求方面取得成功，而且本体可以存储在 Hadoop 存储架构使用的存储模式中。其次，我们需要确保分布式 SPARQL 查询处理器，能够通过底层的 RDF 存储来有效地评估查询（即用户生成的 SPARQL 查询以及评估策略的 SPARQL 查

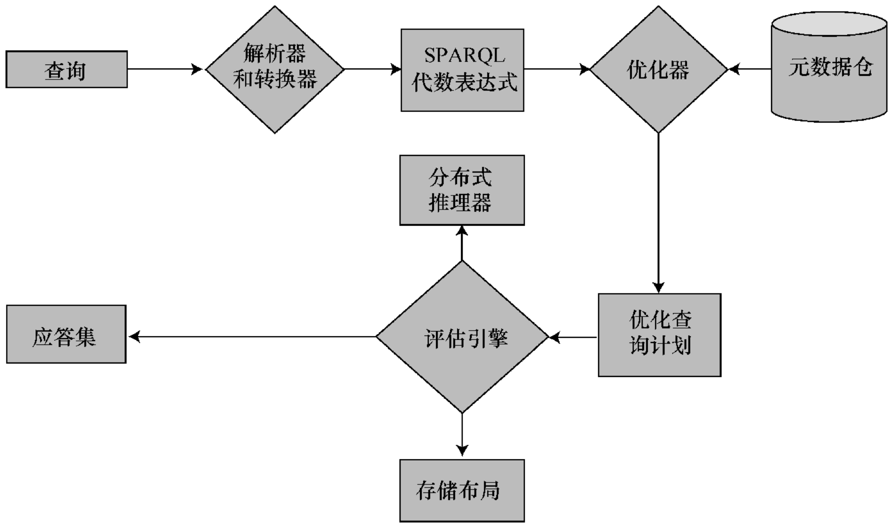


图 27.12　CAISS + + 中的 SPARQL 分布式处理（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 – 26，© Springer，ISBN 978 – 3 – 642 – 30427 – 9）

询）。最后，我们需要对 CAISS + + 进行性能评估，以验证其符合各参与机构的性能要求，并且与 CAISS 相比会带来明显的性能优势。

27. 2. 2. 11　策略规范和实施

CAISS + + 的用户可以使用他们所选择的一种语言（例如 XACML、RDF、REI，等），来指定他们的信息共享策略。这些策略将使用现有或定制的转换器，来转换成合适的 OWL 子语言。我们将为 CAISS 扩展我们的策略引擎，来处理 OWL 特定的策略。此外对于 RDF 策略，我们目前的策略引擎可以处理 OWL 中的策略，其实现是基于角色的访问控制、推理控制和社交网络分析。

27. 2. 3　形式策略分析

我们的框架适用于各种关键任务、高安全需求的应用程序，这些应用程序可能跨越多个不相互信任的组织。为了在这种环境下提供最大的安全保障，对系统的正确性及其执行的策略，建立强有力的保障是非常重要的。最后，为了构建形式的、重要系统属性的机器可检查的证明，我们审查了基础设施的开发过程，并对我们的系统进行了策略分析。虽然机器可检查的证明对于许多大型软件系统来说可能非常困难和耗时，但是我们选择的 SPARQL、RDF 和 OWL 作为查询、本体和策略语言，为在逻辑编程环境中从容地给出此类证明，提供了难得的机会。我们将在 Prolog 中将策略、策略重写算法和安全属性编码作为基于规则的逻辑推导系统，并应用模型检查和定理验证系统（如 ACL2）

来生成机器可检查的证明。我们打算在模型中考虑的属性包括健壮性、透明性、一致性和完整性。我们形式策略分析的结果将推动 CAISS ++的详细设计和实施。据我们所知，以前的工作都没有重点关注过 SPARQL、RDF 和 OWL 的形式策略分析。我们在相关工作中，将讨论对嵌入式参考监视器进行形式策略分析的广泛研究。

27.2.4 实施方法

我们正在用 Java 对 CAISS 进行实现，并且基于一个灵活的设计，我们可以在其中插入和播放多个组件。服务提供商和/或用户具有使用 SPARQL 查询处理器的灵活性，可以像基于 RDF 的策略引擎一样，当作单独组件或组合它们。用于 CAISS 的开源组件将包括 Pellet 推理器以及我们的内部工具，例如 Hadoop / Map/Reduce 框架上的 SPARQL 查询处理器和以云为中心的 RDF 策略引擎。CAISS 将允许我们在基于云的框架上展示基本的 AIS 方案。

对于 CAISS ++的设计，我们将在此使用 Java 作为编程语言。在实施本体论工程的过程中，我们将使用 Protégé 作为本体编辑器，其中包括设计域本体以及上层本体。我们也将评估几个现有的分布式推理算法，比如 WebPIE 和 QueryPIE，来确定满足机构要求的最佳算法。被挑选的算法将在基于 PWL 的安全策略中执行推理。此外，Hadoop 存储架构的设计将基于 Jena 的 SPARQL 数据库（SPARQL Database，SDB）架构，并将具有 Jena SDB 可用的一些功能。SPARQL 查询引擎还将使用 Java 编写的代码。该代码将包含多个模块，包括查询解析和翻译，查询优化和查询执行。查询执行模块将由用于 SPARQL 的各种运算符的 Map/Reduce 作业组成。最后，我们基于 Web 的用户接口将使用如 JBoss、EJB、JSF 等几个组件。

27.3 相关工作

本节将首先提供与项目直接相关的研究概况，然后讨论整体相关工作。本节还将讨论产品或者技术竞争。

27.3.1 我们的相关研究

需注意意到，我们很多的相关研究已在第 13 章、第 22 章、第 23 章中讨论过。我们总结这些研究并讨论其他相关研究。

27.3.1.1 云中的数据存储和检索安全

我们构建了一个基于 Web 的应用程序，它结合了现有的云计算技术，如 Hadoop、开放源码分布式系统、基于 Hadoop 架构的 Hive 数据仓库基础架构、基于 XACML 策略的安全机制，这使协作机构能够安全地存储和检索大量数据（见本章参考文献［THUR10］、［HUSA11］、［UTD1］）。图 27.13 显示了我们系统的体系结构。我们使用 Hive 层和 Hadoop 提供的服务，包括构成 Hadoop 存储层的 Hadoop 分布式文件系统（HDFS）层，并允许在一组节点上存储数据块。我们实现的层包括 Web 应用层、ZQL

解析器层、XACML 策略层和查询重写层。Web 应用层是我们系统向用户提供访问云基础架构的唯一接口。ZQL 解析器（见本章参考文献［ZQL］）层将用户提交的任何查询作为输入，如果查询被成功解析，则会继续执行 XACML 策略评估器。否则，向用户返回错误消息。XACML 策略层用于构建（XACML 策略构建器）和评估（XACML 策略评估）XACML 策略。基本查询重写层，可以重写用户输入的 SQL 查询。Hive 层用于管理存储在基础 Hadoop HDFS（见本章参考文献［THUS09］）中的关系数据。另外，我们还在混合云中设计并实现了安全存储和查询处理（见本章参考文献［KHAD11］）。

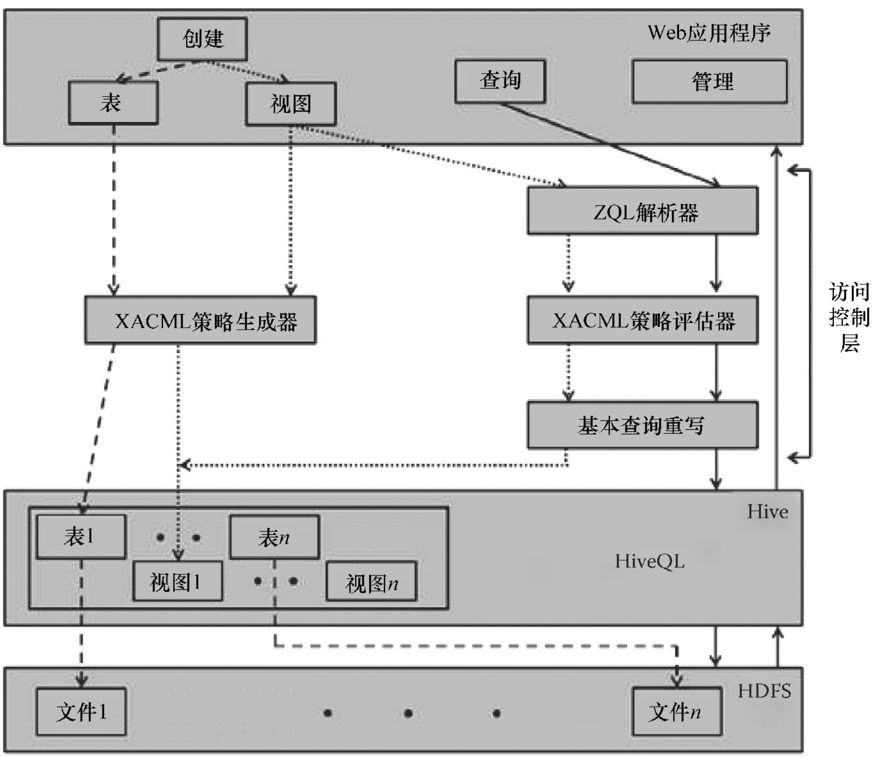


图 27.13　基于 HIVE 保证云查询处理（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics – Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1 – 26，© Springer，ISBN 978 – 3 – 642 – 30427 – 9）

27. 3. 1. 2　云上的 SPARQL 查询处理安全

我们已经开发了一个框架，来查询存储在 Hadoop 中的 RDF 数据，如图 27. 14 所示。我们用 Pellet 推理器在不同的阶段推理。我们也使用与 Hadoop Map/Reduce 功能相结合的 Pellet 库进行实时查询推理。我们的 RDF 查询处理由两个主要步骤组成：预处理和查询优化和执行。

27.3.1.2.1　预处理

要对 RDF 数据执行 SPARQL 查询，要先执行数据预处理步骤，并将预处理数据存储在 HDFS 中。写一个单独的 Map/Reduce 任务来执行，并将 RDF / XML 数据转换为 N－Triples，同时也是为了前缀的生成。这里的存储策略是基于谓词分割的（见本章参考文献［HUSA11］）。

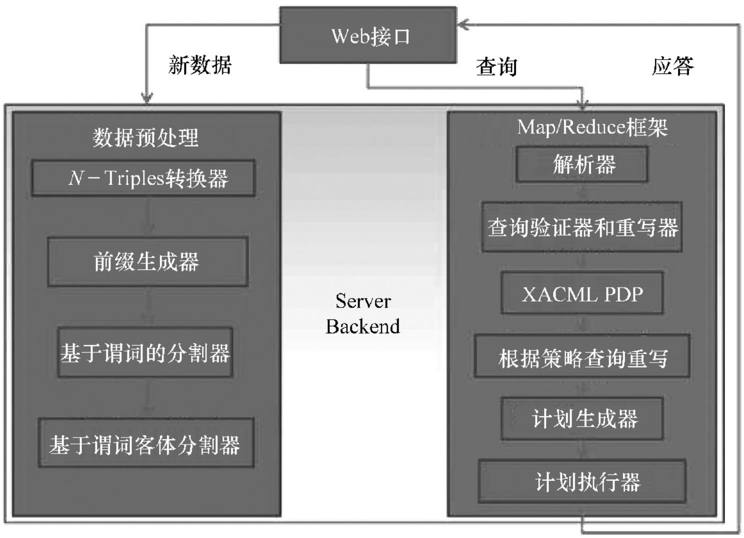


图 27.14　基于 SPARQL 的云查询处理安全（得到了 Springer 这一科学和商业媒体的允许，内容来自：Khadilkar，V.，Rachapalli，J.，Cadenhead，T.，Kantarcioglu，M.，Hamlen，K. W.，Khan，L. 和 Husain，M. F. Media from Lecture Notes in Computer Science，Proceedings of Intelligence and Security Informatics－Pacific Asia Workshop，PAISI 2012，Kuala Lumpur，Malaysia，7299，2012，p. 1－26，© Springer，ISBN 978－3－642－30427－9）

27.3.1.2.2　查询执行和优化

我们已经为 Hadoop 开发了一个 SPARQL 查询执行和优化模块。由于我们的存储策略是基于谓词分割的。所以，第一，我们要检查查询中存在的谓词。第二，我们要检查输入文件与谓词匹配的子集。第三，通常在 SPARQL 查询中具有许多连接，并且所有这些连接可能无法在单个 map－reduce 作业中执行。因此，我们已经开发了一种算法，来决定每种查询所需的作业数量。作为优化的一部分，我们采用贪婪算法和基于成本优化来减少查询处理时间。我们还开发了一个基于 XACML 的集中式策略引擎，将在云上执行联合 RDF 查询处理。执行策略的细节在本章参考文献［HUSA11］、［KHAL10］、［HAML10b］中给出。

27.3.1.3　RDF 策略引擎

在我们之前的工作中（见本章参考文献［CADE11a］），已经为 RDF 数据开发了一个策略引擎来处理基于 RDF 的访问控制策略。策略引擎设计具有以下功能：可扩展性、高效性和互操作性。这个框架（见图 27.15）可以用来执行各种策略，包括访问控制策略和修订策略，它还可以用作测试平台，评估针对 RDF 数据的不同策略集，并可以图

形方式查看结果。我们的框架提出了接受高层次策略的接口，然后转换成所需的格式。它根据用户输入的查询，返回一组经过用户定义的策略约束修整的回复。这个框架是使用模块化方法构建的，因此它是非常灵活的，因为大多数模块可以被另一个模块扩展或替换。例如，实施自主访问控制（Discretionary Access Control，DAC）的策略模块，可以被 RBAC 模块完全替换，或者我们可以决定执行基于广义修正模型的所有约束。应该注意的是，我们的策略引擎还处理 OWL 中指定的基于角色的访问控制策略 SWRL（见本章参考文献［CADE10］）。另外，它处理 OWL 中指定的某些策略以进行推理控制，例如一个基于关联的策略，访问实体集合被拒绝并且逻辑策略 A 包含 B。如果对 B 的访问被拒绝，那么对 A 的访问也应被拒绝（见本章参考文献［CADE10］、［CADE11b］、［CARM09］）。我们策略引擎的这种功能，将在我们设计和实现的 CAISS + + 中有用，其中的信息也将在多个云中共享。

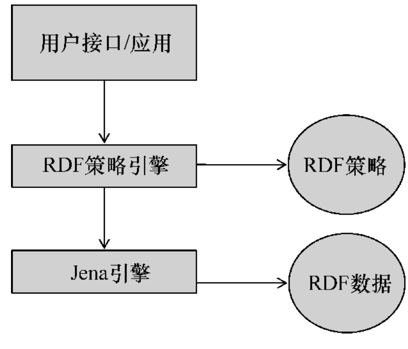


图 27.15　RDF 策略引擎

27.3.1.4　ASI 原型

在 UTD，我们已经为 AIS 开发了多个系统。特别的，我们开发了一个基于 XACML 的策略引擎，它在关系数据库之上运行，并且展示了医学数据的共享（模拟）（见本章参考文献［THUR08］）。在这种实验中，我们指定了 XACML 中的策略，并且将数据存储在多个 Oracle 数据库中。当一个组织向另一个组织请求数据时，会对这些策略进行检查并授权数据的发布。另外，我们也在共享信息时，对执行策略的数据量进行了模拟研究。我们还对基于激励的信息共享，进行了模拟研究（见本章参考文献［KANT10］）。此外，我们已经在信息共享方案中，检查了基于风险的访问控制（见本章参考文献［CELI07］）。除了访问控制策略，我们还指定了不同类型的策略，包括需要共享的策略和信任策略（例如 A 和 B 共享数据，B 不与 C 共享数据）。需要注意的是，9/11 的委员会报告要求从更严格的需求获知转移到限制性较少的需要共享策略。这些策略是支持有关 AIS 职责规范的关键。我们在附录 D 中讨论了关于 AIS 的工作。

27.3.1.5　形式策略分析

通过将高级别安全策略规范和系统模型，降低到其二进制实现的符号和操作语义的级别，我们之前的工作，已经开发出机器可认证的安全实施各种复杂软件系统的形式化机制，包括在 .NET（见本章参考文献［HAML06b］）、ActionScript（见本章参考文献［SRID10］）、Java（见本章参考文献［JONE10］）和本机代码（见本章参考文献［HAML10b］）实现的各种系统。在二进制级别工作中提供了极高的形式保证，因为它允许生成任务关键软件组件的工具链来鉴别不信任。由链产生的二进制码可以直接认证。这种策略对于 CAISS + + 来说非常出色，因为 XACML 和 OWL 等数据安全性规范语言，借助识别 XML 的系统 API，轻松地转换为字节码语言的二进制形式，比如 Java 字节码语言。我们过去的工作已经应用了二进制检测（例如嵌入式参考监视器）和二进制检查（见本章参考文献［HAML06b］）与模型检查（见本章参考文献［SRID10］）

的结合，并且使用自动化定理证明（例如，通过 ACL2）以实现在这种域中二进制软件的全自动机器认证。

27.3.2 总体相关研究

虽然现在有一些相关的努力，但是没有一个为云中使用 AIS 提供解决方案，也没有实施这样的形式策略分析。

27.3.2.1 云中的数据存储和检索安全

云的安全性引起了最新关注（见本章参考文献［TALB09］）。我们已经报告了在基础设施层面上实施的一些成就（见本章参考文献［OMAL09］）。这样的开发成就是保护云基础设施的重要一步，但只是在开始阶段。我们系统的目标是在 Hadoop（见本章参考文献［UTD1］）的安全性之上添加另一层安全性。一旦 Hadoop 的安全性变得强大，便会加强我们系统的有效性。亚马逊和微软已经为云计算（AMAZ09）（见本章参考文献［MARS10］）进行了类似的工作。然而，这项工作属于公有领域，而我们的系统是专为私有云基础设施而设计的。这个区别因素使得我们的基础设施要“信任”公共基础设施，数据就必须以加密格式存储。

27.3.2.2 SPARQL 查询处理器

据报道在 SPARQL 查询处理方面目前已经有了些许成就。这些包括 BioMANTA（见本章参考文献［BIOM11］）和 SHPRD（见本章参考文献［SHPR11］）。BioMANTA 建议扩展 RDF 分子（见本章参考文献［DING05］），并实施基于 Map/Reduce 的分子存储。他们使用 Map/Reduce 来回答查询，并已经查询了最多 400 万的三元组。我们的工作在以下方面有所不同：首先，我们查询了 10 亿个三元组。其次，为了提高 RDF 数据的查询执行性能，我们设计了一个存储模式。据我们所知，我们是第一个在 HDFS 中提出 RDF 数据的存储模式的，并使用 Map/Reduce 作业确定算法来回答一个 SPAEQL 查询。SHPRD 是使用 Hadoop Cloudera 分布式的 RDF 三元组存储。该项目显示的初步结果表明了 Hadoop 具有提高 RDF 数据集可扩展性的能力。但是，SHPRD 仅仅使用三元组存储模式来存储数据。它没有查询规划或重新排序，而且它的查询处理器不会最小化 Hadoop 作业的数量。另外，在安全策略上也没有一项成就。

27.3.2.3 基于 RDF 的策略引擎

目前已有针对 RDF 存储的执行策略的相关研究。这些研究包括本章参考文献［CARM04］中提到的工作，使用 RDF 进行策略规范和实现。另外，策略通常在 RDF 中编写。在本章参考文献［JAIN06］中，作者对 RDF 提出了一个访问控制模型。他们的模型是基于 RDF 数据语义的，并且结合 RDF 和 RDF 模式（RDF Schema，RDFS）的需求。在这里，在资源层提供保护，并为其框架增加了粒度。在 RDF/OWL 上的其他框架执行策略包括本章参考文献［USZO04］、［KAGA02］。本章参考文献［USZO04］描述了 KAoS，一种使用 OWL 的策略和域服务的框架，用来表示策略和域。本章参考文献［KAGA02］介绍了 REI，一种策略框架，它是灵活的并且允许不同种类的策略被陈述。最近也有人提出了 REI 的扩展（见本章参考文献［KHAN10］）。这种策略规范语言允许用户对 RDF、DAML + OIL、OWL 的域特定本体开发有声明性的策略。本章参考文献

［REDD05］中的作者还引入一个原型 RAP，用于实施具有集成功能和访问控制的 RDF 存储。但是，这些框架并没有解决 RDF 存储可能会变得非常大，或策略不与数据同步扩展的情况。在 IARPA 资助的项目下，我们还开发了处理非常大 RDF 图形的技术（见本章参考文献［UTD2］）。

27.3.2.4 Hadoop 存储架构

人们对 RDF 数据大规模分布式存储和检索技术有很大兴趣。关于数据并行处理框架的理论设计的工作由 Castagna 等人完成（见本章参考文献［CAST09］）。这项工作主张使用具有不同级别粒度的数据分布模型，例如三元组级别、图形级别和数据集级别。在分布式模型上的查询（包含分布式数据的机器上的）被分成一组子查询。所有子查询的结果将被合并，以完整的结果返回给用户应用程序。这个理论概念的几个实现存在于研究界，这些努力包括 Choi 等人（见本章参考文献［CHOI09］）和 Abraham 等人（见本章参考文献［ABRA10］）的工作。一项已经用于存储和检索 RDF 数据的专门技术，使用了 P2P 系统（见本章参考文献［ABER04］、［CAI04］、［HART07］、［VALL06］）。然而，这种系统存在一些缺陷，因为 P2P 系统需要具有超级对等体，来存储关于对等体之间的 RDF 数据分布的信息。另一个缺点是需要将 SPARQL 查询联合到网络中的每个对等体。

27.3.2.5 分布式推理

InteGrail 系统使用分布式推理，其愿景是塑造未来的欧洲铁路组织（见本章参考文献［INTE09］）。在本章参考文献［URBA09］中，作者已经展示了基于 Map/Reduce 的 RDFS 推理上实现的可扩展性，可以在不到 2 小时的时间内，从现实世界的数据集中推算出 300 亿个三元组，引发的输入和输出吞吐量分别为每秒 123.000 个三元组和每秒 327 万个三元组。他们为 Map/Reduce 中的 RDFS 规则集进行了一些重要的优化，并且使用几个现实世界的数据集，评估了在 64 个计算节点的集群上实现的可扩展性。

27.3.2.6 访问控制和策略本体建模

已经有一些人，尝试使用语义 Web 技术来建模访问控制和策略模型。在本章参考文献［CIRI07］中，作者已经展示了如何使用 OWL 和描述逻辑来构建访问控制系统。他们开发了一个高级 OWL－DL 本体，表达了基于角色的访问控制系统的元素，并已经构建了一个能够捕获示例场景特征的特定域本体。最后，他们考虑到策略的齿列和访问控制决策中的属性，加入了这两个比较老的组件。在本章参考文献［REUL10］中，作者首先提出了一种基于 DOGMA 的安全策略本体，这是一个形式本体工程框架。本体论涵盖了安全策略的核心要素（即条件、行动、资源），并且可以轻松地扩展以表示特定的安全策略，如访问控制策略。在本章参考文献［ANDE09］中，作者提出了用一种本体论动机的方法来应对信息系统的多级访问控制和多级别来源。

27.3.3 商业发展

27.3.3.1 RDF 处理引擎

研究用和商业化的 RDF 处理引擎包括惠普实验室的 Jena，BigOWLIM 和 RDF－3X。

虽然这其中有些存储模式和查询处理机制是专有的，但它们都是基于 RDF 数据中某种类型的索引策略。但是，只有少数的工具使用云中心架构来处理 RDF 数据，此外，这些工具不可以扩展成大量数目的三元组。相比之下，我们在 CAISS ++中的查询处理器将构建为行星级别规模的 RDF 处理引擎，支持所有 SPARQL 运算符，并为 SPARQL 查询提供优化的执行策略，并可以扩展到数十亿个三元组。

27.3.3.2 基于 Web 的语义安全策略引擎

如 27.3.2 节所述，目前关于基于 Web 语义策略规范和实施的工作，没有解决面向大型数据的策略生成和实施，以及支持大量用户的问题。

27.3.3.3 云

据我们所知，对于云中心 AIS 并不存在明显的商业竞争。我们采用模块化的方法来创建我们的工具，可以分别迭代地处理每个组件（策略引擎、存储架构和查询处理器）。由于采用基于组件的方法，我们将能够适应所使用平台的变化（例如 Hadoop、RDF、OWL 和 SPARQL），而不必依赖给定平台的特定功能。

27.4 总结和展望

本章介绍了我们基于云信息共享而设计的一个系统，叫作 CAISS。CAISS 利用了我们开发的多种技术和开源技术。我们也描述了一种理想的基于云计算的信息共享保障系统设计，叫作 CAISS++。

我们已经实现了 CAISS 的几个版本。在 CAISS 的第一个版本实施中，我们利用 SPARQL 查询处理器与 XACML 指定的策略，这或多或少是第 23 章中描述的系统。第二个版本实施中，我们利用 RDF 中指定策略的原型，在云中开发策略引擎并将其与数据引擎集成。这个系统将在第 28 章讨论（另见本章参考文献［CADE12a］和［CADE12b］）。未来我们将继续通过实施更复杂的策略，来增强我们的原型。未进行分析的策略包括访问控制策略和信息共享策略。我们将对策略的执行实施形式进行分析。我们的最终目标是实现 CAISS++。

参考文献

[ABRA10] Brazier, A., C. Navarro, and A. Piazza. Distributed storage and querying techniques for a semantic web of scientific workflow provenance. *Proceedings IEEE SCC*, Miami, FL, 2010.

[ABER04] Aberer, K. P., P. Cudré-Mauroux, M. Hauswirth, and T. Van Pelt. GridVine: Building Internet-scale semantic overlay networks. *International Semantic Web Conference*, Hiroshima, Japan, 2004.

[AMAZ09] Amazon Web Services: Overview of Security Processes. http://awsmedia.s3.amazonaws.com/pdf/AWSSecurityWhitepaper.pdf

[ANDE09] Andersen, B. and F. Neuhaus. An ontological approach to information access control and provenance. *Proceedings of Ontology for the Intelligence Community*, Fairfax, VA, October 2009.

[AWAD10] Khan, A. and B. M. Thuraisingham. Policy enforcement system for inter-organizational data sharing. *Journal of Information Security and Privacy*, 4(3): 22–39, 2010.

[BIOM11] BioMANTA: Modelling and analysis of biological network activity. http://www.itee.uq.edu.au/reresearch/projects/biomanta.

[CADE10] Cadenhead, T., M. Kantarcioglu, and B. M. Thuraisingham. Scalable and efficient reasoning for enforcing role-based access control. *Proceedings of Data and Applications Security and Privacy XXIV, 24th Annual IFIP Working Group 11.3 Working Conference*, Rome, Italy, p. 209–224, 2010.

[CADE11a] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. M. Thuraisingham. Transforming provenance using redaction. *Proceedings of ACM Symposium on Access Control Models and Technologies*, Innsbruck, Austria, p. 93–102, 2011.

[CADE11b] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. M. Thuraisingham. A language for provenance access control. *Proceedings of ACM Conference on Data Application Security and Privacy*, San Antonio, TX, 133–144, 2011.

[CADE12a] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. M. Thuraisingham. A cloud-based RDF policy engine for assured information sharing. *Proceedings of ACM Symposium on Access Control Models and Technologies*, Newark, NJ, 113–116, 2012.

[CADE12b] Cadenhead, T., M. Kantarcioglu, V. Khadilkar, and B. M. Thuraisingham. Design and implementation of a cloud-based assured information sharing system. *Proc. of Intl. Conf. on Mathematical Methods, Models and Architectures for Computer Network Security*, St. Petersburg, Russia, 36–50, 2012.

[CARM04] Carminati, B., E. Ferrari, and B. M. Thuraisingham. Using RDF for policy specification and enforcement. *Proc. of Intl. Workshop on Database and Expert Systems Applications*, Zaragoza, Spain, p. 163–167, 2004.

[CARM09] Carminati, B., E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. M. Thuraisingham. A semantic web based framework for social network access control. *Proc. of ACM Symposium on Access Control Models and Technologies*, Stresa, Italy, 177–186, 2009.

[CAI04] Cai, M. and M. Frank. RDFPeers: A scalable distributed RDF repository based on a structured peer-to-peer network. *Proceedings ACM World Wide Web Conference*, New York, NY, 2004.

[CELI07] Celikel, E., M. Kantarcioglu, and B. M. Thuraisingham, ElisaBertino. Managing risks in RBAC employed distributed environments. *On the Move to Meaningful Internet Systems*, Vilamoura, Portugal, (2): 2007.

[CAST09] Castagna, P., A. Seaborne, and C. Dollin. A parallel processing framework for RDF design and issues. Technical report, HP Laboratories, HPL-2009-346, 2009.

[CHOI09] Choi, H., J. Son, Y. Cho, M. Sung, and Y. Chung. SPIDER: A system for scalable, parallel/distributed evaluation of large-scale RDF data. *Proceedings ACM Conference on Information and Knowledge Management (CIKM)*, Hong Kong, China, 2087–2088, 2009.

[CIRI07] Cirio, L., I. Cruz, and R. Tamassia. A role and attribute based access control system using semantic web technologies. *IFIP Workshop on Semantic Web and Web Semantics*, Vilamoura, Algarve, Portugal, 2007.

[DING05] Ding, L., T. Finin, Y. Peng, P. da Silva, and D. Mcguinness. Tracking RDF graph provenance using RDF molecules. *Proc. International Semantic Web Conference*, Galway, Ireland, 2005.

[DoD] DoD Information Enterprise Strategic Plan, 2010-2012, http://cio-nii.defense.gov/docs/DodIESP-r16.pdf

[DoD07] Department of Defense Information Sharing Strategy, 2007.

[DoD09] DoD mbraced Cloud Computing, http://www.defensemarket.com/?p=67

[FINI09] Finin, T., A. Joshi, H. Kargupta, Y. Yesha, J. Sachs, E. Bertino, Li et al., Assured information sharing life cycle. *Proc. Intelligence and Security Informatics*, 2009.

[GUO05] Guo, Y., J. Heflin, and Z. Pan. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics*, 3(2-3), 158–182, 2005.

[HAML06a] Hamlen, K., G. Morrisett, and F. Schneider. Computability classes for enforcement mechanisms. *ACM Transactions on Programming Languages and Systems*, 28(1): 175–205, 2006.

[HAML06b] Hamlen, K., G. Morrisett, and F. Schneider. Certified in-lined reference monitoring on.NET. *Proc. ACM Workshop on Prog. Lang. and Analysis for Security*, pp. 7–16, Ottawa, Canada, 2006.

[HAML10b] Hamlen, K., V. Mohan, and R. Wartell. Reining in Windows API abuses with in-lined reference monitors. Tech. Rep. UTDCS-18–10, Computer Science Dept., University of Texas at Dallas, 2010.

[HART07] Harth, A., J. Umbrich, A. Hogan, and S. Decker. YARS2: A federated repository for searching and querying graph structured data. *Intl. Semantic Web Conference,* Busan, Korea, 2007.

[HUSA11] Husain, M., J. McGlothlin, M. Masud, L. Khan, and B. M. Thuraisingham. Heuristics-based query processing for large RDF graphs using cloud computing. *IEEE Trans. Knowl. Data Eng.*, 23, 1312–1327, 2011.

[INTE09] Distributed reasoning: Seamless integration and processing of distributed knowledge. http://www.integrail.eu/documents/fs04.pdf.

[JAIN06] Jain, A. and C. Farkas. Secure resource description framework: An access control model. *ACM Symposium on Access Control Models and Technologies,* Lake Tahoe, CA, 2006.

[JONE10] Jones, M. and K. Hamlen. Disambiguating aspect-oriented security policies. *Proc. 9th Int. Conf. Aspect-Oriented Software Development*, Rennes and St. Malo, France, pp. 193–204, 2010.

[JONE11] Jones, M. and K. Hamlen. A service-oriented approach to mobile code security. *Proc. 8th Int. Conf. Mobile Web Information Systems,* (MobiWIS) Niagara Falls, Ontario, Canada, 2011.

[KAGA02] Kagal, L. REI., A policy language for the project, http://www.hpl.hp.com/techreports/2002/HPL-2002-270.pdf. 2002

[KANT10] Kantarcioglu, M. Incentive-based assured information sharing, *AFOSR MURI Review*, October 2010.

[KHAD11] Khadilkar, V., M. Kantarcioglu, and B. M. Thuraisingham. Mehrotra: Secure data processing in a hybrid cloud. *Proc. Computing Research Repository/1105.1982*, May 2011.

[KHAL10] Khaled, A., M. Husain, L. Khan, K. Hamlen, and B. M. Thuraisingham. A token-based access control system for RDF data in the clouds. *CloudCom2010*, Indianapolis, IN.

[KHAN10] Khandelwal, A., J. Bao, L. Kagal, I. Jacobi, L. Ding, and J. Hendler. Analyzing the AIR language: A semantic web (production) rule language. *Intl. Conf. on Web Reasoning and Rule Systems*, Bressanone, Brixen, Italy, 58–72, 2010.

[MARS10] Marshall, A., M. Howard, G. Bugher, and B. Harden. Security best practices in developing Windows Azure applications. *Microsoft Corp*. 2010.

[MEWM08] Newman, A., J. Hunter, Y. Li, C. Bouton, and M. Davis. A scale-out RDF molecule store for distributed processing of biomedical data. *Semantic Web for Health Care and Life Sciences Workshop, World Wide Web Conference*, Beijing, China, 2008.

[NSA11] *Information Week*, NSA pursues intelligence-sharing architecture, http://www.informationweek.com/news/government/cloud-saas/229401646. 2011.

[OMAL09] O'Malley, D., K. Zhang, S. Radia, R. Marti, and C. Harrell. Hadoop Security Design. http://techcat.org/wp-content/uploads/2013/04/hadoop-security-design.pdf. 2009.

[RAO08] Rao, P., D. Lin, E. Bertino, N. Li, and J. Lobo. EXAM: An environment for access control policy analysis and management. *IEEE Workshop on Policies for Distributed Systems and Networks (POLICY)*, Palisades, NY, 2008.

[REDD05] Reddivari, P., A. Joshi, and T. Finin. Policy-based access control for an RDF store. *Policy Management for the Web, IJCAI Workshop*, Chiba, Japan, 2005.

[REUL10] Reul, Q., G. Zhao, and R. Meersman. Ontology-based access control policy interoperability. *Proc. 1st Conference on Mobility, Individualisation, Socialisation and Connectivity*, MISC, London, UK, 2010.

[SHPR11] SHPRD. http://www.cloudera.com/blog/2010/03/how-raytheonresearchers-are-using-hadoop-to-build-a-scalable-distributed-triplestore.

[SRID10] Sridhar, M. and R. Hamlen. Model-checking in-lined reference monitors. *Proc. 11th Int. Conf. on Verification, Model Checking, and Abstract Interpretation*, Madrid, Spain, pp. 312–327, 2010.

[TALB09] Talbot, D. How secure is cloud computing, 2009, http://www.technologyreview.com/computing/23951/

[THUR08] Thuraisingham, B. M., H. Kumar, and L. Khan. Design and implementation of a framework for assured information sharing across organizational boundaries. *Journal of Information Security and Privacy*, 2(4): 67–90, 2008.

[THUR10] Thuraisingham, B. M., V. Khadilkar, A., Gupta, M. Kantarcioglu, and Khan. Secure data storage and retrieval in the cloud. *CollaborateCom*, Chicago, IL, 2010.

[THUR11] Thuraisingham, B. M. and V. Khadilkar. Assured information sharing in the cloud. *UTD Tech. Report*. Sept. 2011.

[THUS09] Thusoo, A., J. Sharma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy. Hive—A warehousing solution over a map-reduce framework. *Proceedings of VLDB Endowment*, 2009.

[THUR12] Thuraisingham, B. M., V. Khadilkar, J. Rachapalli, T. Cadenhead, M. Kantarcioglu, K. W. Hamlen, L. Khan, and M. F. Husain. Cloud-centric assured information sharing. *PAISI*, Kuala Lumpur, Malaysia, 2012, 1–26.

[URBA09] Urbani. Scalable Distributed Reasoning using MapReduce. http://www.few.vu.nl/~jui200/papers/ISWC09-Urbani.pdf.

[USZO04] Uszok, Bradshaw, Johnson, Jeffers, Tate, Dalton, and Aitken. KAoS policy management for semantic web services. *IEEE Intelligent Systems*, 19(4): 32–41, 2004.

[UTD1] UTD Secure Cloud Repository, http://cs.utdallas.edu/secure-cloud-repository/

[UTD2] UTD Semantic Web Repository, http://cs.utdallas.edu/semanticweb/

[VALL06] Valle, E., A. Turati, and A. Ghioni. AGE: A distributed infrastructure for fostering RDF-based interoperability. *Proceedings Distributed Applications and Inter-Operable Systems (DAIS)*, Bologna, Italy, 2006.

[ZQL] Zql: A Java SQL parser. http://www.gibello.com/code/zql/

第 28 章　基于语义云信息共享保障系统的设计与实现

28.1　概述

云计算模式可以安全有效地共享大量数据。此外，云计算的出现以及软件即服务（Software as a Service，SaaS）模式的持续发展，已经对作为云中服务的保障信息共享（AIS）的需求持续上升。为了满足联盟组织以云为中心的 AIS 需求，需要开发一个在云中运行的 AIS 框架。据我们所知，目前还没有这样的系统存在。第 27 章中，我们描述了一个系统叫作 CAISS 的设计：云中心信息共享保障系统（Cloud Centric Assured Information Sharing System，CAISS），它利用了我们之前内部设计的技术组件和开源工具。CAISS 包括两个部分：一个是云中心策略管理器，执行 RDF 指定的策略（见本章参考文献［KLYN04］），另一个是云中心数据管理器，将存储和管理 RDF 指定的数据。RDF 数据管理器本质上是 SPARQL 的查询引擎，SPARQL 是语义 Web 数据广泛使用的用于查询 RDF 数据的语言。RDF 是语义 Web 语言，它被认为比基于 XML 策略语言对策略更具表达力和推理力。另外，我们的策略管理器和数据管理器将无缝集成，这是由于它们都管理 RDF 数据。

上一章讨论的系统（例如 CAISS，CAISS + +）正在设计阶段，本章，我们在语义云中描述设计和实施的细节。也就是说，我们使用语义 Web 技术提供基于云计算的语义 Web 服务。这种语义 Web 服务实现信息共享。这或多或少是第 27 章中描述的 CAISS 系统。我们基本上开发了一个全面的 AIS 框架在云中无缝运行。我们的框架包括 3 层结构，包含一个用户接口层、一个策略引擎层和一个将多个数据源集成到云中的数据连接层。据我们所知，这是第一个在云中运行的 AIS 框架。我们在 28.2 节中将描述我们系统的设计和实施的细节。尤其是，系统结构、系统结构操作、系统结构模块以及系统结构使用都将被讨论。本章最后在 28.3 节中讨论关于未来的工作。

28.2　架构

28.2.1　简介

我们的策略引擎架构由 RDF 配置文件驱动，配置文件对策略引擎的逻辑、它们的

使用、用户接口布局和可定制的参数进行编码，并使用可靠的数据连接将可参引的 Uris 映射到数据存储。我们的策略引擎框架可以作为增强 RDBMS（关系数据库管理系统）和基于云计算系统的安全性关键推动因素。RDBMS 是以原子性、并发性和耐久性为核心开发的，但通常对访问控制的支持有限。云存储层允许各机构存储，并扩展具有对 RDF 资源有良好控制权限的策略。云的发展考虑到了可扩展性和可用性，但安全问题被忽略。我们的策略引擎可以在 RDBMS 系统中的一个入口完成配置策略，来支持基于云的安全策略。我们首先提出框架的配置，然后定义我们架构中的各层。最后，我们描述我们实施中的新特点。图 28. 1 说明了我们的架构，图 28. 2 为我们的配置框架。

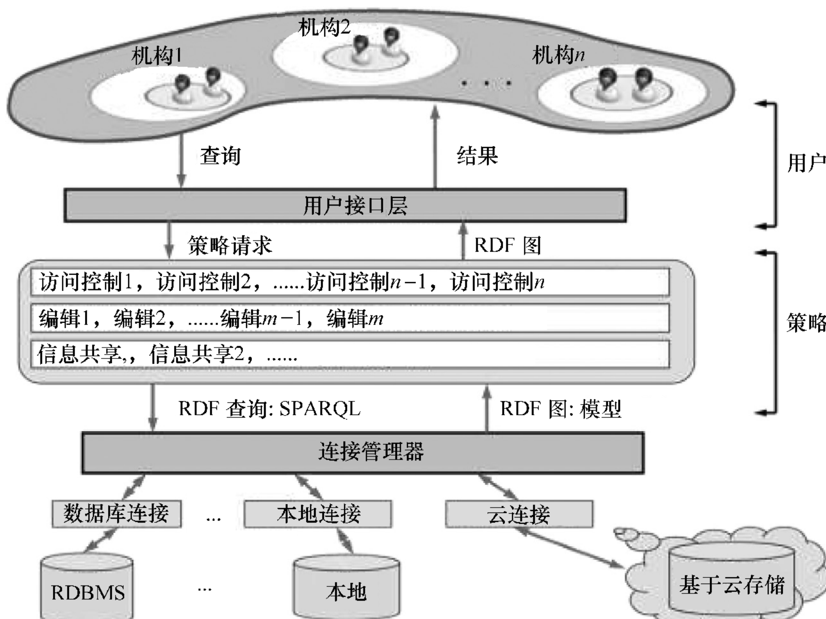


图 28. 1　架构（得到了 Springer 这一科学和商业媒体的允许，内容来自：Cadenhead，T. ，Kantarcioglu，M. 和 Khadilkar，V. ，Design and Implementation fo a Cloud－Based Assured Information Sharing System ，Media from Lecture Notes in Computer Science，Proceedings of Computer Network Security－6th International Conference on Mathematical Methods，Models and Architectures for Computer Network Security，MMM－ACNS 2012，St. Petersburg，Russia，7531，2012，p. 36－50，© Springer）

28. 2. 2　框架配置

松散耦合系统给我们的 RDF 策略引擎框架提供了简单的配置和灵活性。每个组件都通过采用 RDF 文档从其他组件中抽象出来，RDF 文档由代理机构对策略的偏好或对数据存储的数据连接组成。此外，松散耦合 Web 前端促使策略框架更容易维护和可重

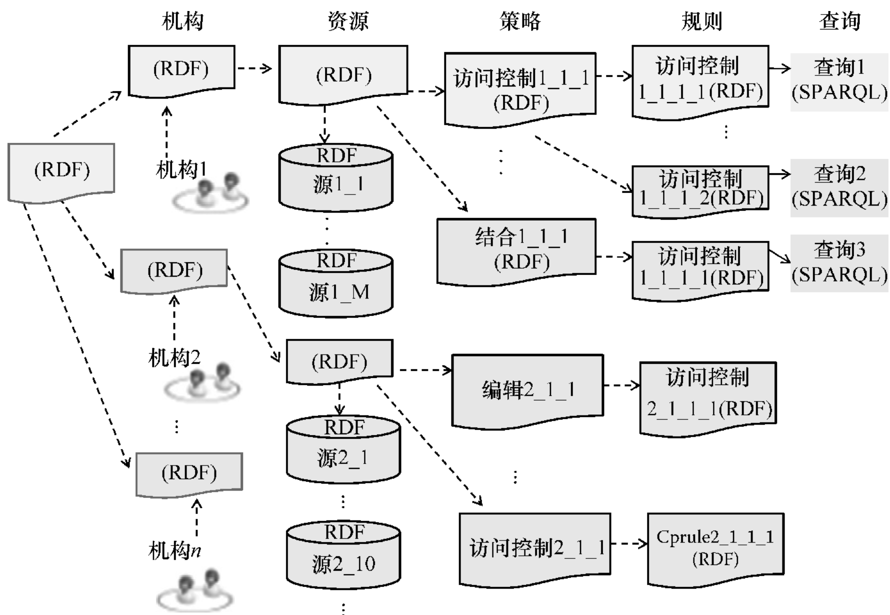


图 28.2 配置概述（得到了 Springer 这一科学和商业媒体的允许，内容来自：Cadenhead，T.，Kantarcioglu，M. 和 Khadilkar，V.，Design and Implementation fo a Cloud－Based Assured Information Sharing System ，Media from Lecture Notes in Computer Science，Proceedings of Computer Network Security－6th International Conference on Mathematical Methods，Models and Architectures for Computer Network Security，MMM－ACNS 2012，St. Petersburg，Russia，7531，2012，p. 36－50，© Springer）

用，因为适配器模式将 Web 接口（和通信）映射到其他层。抽象隐藏了来自机构的策略引擎管理器和数据管理器的真正实施和复杂性。因此这允许机构用任何语言来表示他们指定的策略，比如 XML、RDF、或 REI（见本章参考文献［KAGA02］）。另外，适配器也将高级策略规范隐藏到策略实施中。

28.2.3 架构中的模块

我们的系统架构包括 3 个层：前端，我们有用户接口；中间层，由策略引擎逻辑组成；后端，我们有数据存储。接下来，我们将讨论这些模块。

28.2.3.1 用户接口层

为了与我们架构一对一交互，在策略层之上建立了基于 Web 的用户接口。客户端和开源 Web 技术简化了用户在 Web 网页、底层策略和数据层的交互。这种整合有许多优点。策略框架在分布式环境中允许具有较好的地域分布。因此，机构和用户都有流动性。Web 接口需要用户创建一个账户（注册）并且选择唯一的凭证，然后由用户将其用于向策略框架标识自己。基于表单的身份验证模式以及挑战性响应测试，将合法用户与机器人（可能会成为正常用户）区别开来。然后，向合法用户提供一个查询接口，

并允许他们在被授权后提交 APSRQL 查询。注意到 SPARQL（见本章参考文献［PRUD06］）是 RDF 的查询语言，并且被用于从三元组存储中检索数据。SPARQL 查询被验证后传给策略引擎层，并返回一个 RDF 结果图显示在网页上。

28.2.3.1.1　用户注册

用户注册向用户提供通过网络注册表向系统注册的机会。注册表获取用户的姓名、密码和其他关于用户的元数据。元数据可以是一个代理，用户是其中的一部分，或者是用户执行的将用户的凭证映射到角色的数据。接下来的 RDF 图展示了一个用户配置文件的内容。RDF 图中的最后一个三元组包含另一个 RDF 图的可参引的 URI，然后它包含用户被允许查询的实际资源中的可参引的 URI 列表。

```
<http://policy.org/agency/pol#users>
   pol:user <http://policy.org/agency/pol#user1>.

# resources
<http://policy.org/agency/pol#user1>
  pol:name "user1";
  pol:passwd "_:b1";
    pol:organization <http://policy.org/agency/pol#Agency1>;
    pol:resourcelist <http://example/users/resources/user1>.
```

28.2.3.1.2　机构注册

机构注册包含一组网页，每个网页都是前一个的子网页。这个过程从一个机构注册信息来开始描述自己。第一，机构注册关于自己的重要元数据。元数据是 RDF 文件，可以被用来介绍一个机构给其他人，因此，应该是自我描述。这个元数据中的一些示例三元组可以表明一个机构的名字、地址、行业和联系方式等。第二，机构记录其资源。资源有唯一的 URI 并且对于机构的 RDF 文件可参引，其中包含该机构的敏感数据和非敏感数据。这是通常存储在关系数据库中的信息，但现在要迁移到云中。第三，一个机构定义其资源策略。机构可以在策略引擎层支持的多种策略中选择。策略的例子包括：访问控制、编辑、信息共享等。第四，机构为策略描述多种策略规则。需注意机构可能使用访问控制保护其资源。但是，机构选择特殊策略可能需要不止一个规则。比如，一个访问控制规则可能指定一个肯定的授权，而在相同资源上其他规则可能指定一个否定的授权。最后，一个机构还要指定查询。在数据存储上写入策略规则是一个非常流行的技术（比如 SPARQL 查询）。机构可以在它的此类规则中指定配置文档，用来说明查询是否被具体化。一个具体化的查询可以加快策略执行速度，而一个未具体化的查询要实时刷新结果集。

28.2.3.2　策略引擎

策略引擎首先根据存储的数据资源评估用户的查询（可以是传统数据或元数据）。数据资源的特征是统一资源标识符（URI）。策略层使用工厂对象来创建基础策略。工厂通过一致的接口公开一项策略，因此使扩展我们的策略引擎来支持其他类型的策略变得更容易。我们目前支持访问控制、编辑和信息共享策略。为了支持传统策略，我们使用 SPARQL 查询来定义资源上的视图，其中视图可以和正负授权或子图替换过程中的目

标相关联。一个重要的元数据是数据源，它记录了一段数据项的历史。但是，数据源采用定向非循环图（Directed Acyclic Graph，DAG）的结构，所以需要自己的策略（见本章参考文献［BRAU08］）。因此我们支持使用正则表达式，从 SPARQL 查询中提取访问控制策略（见本章参考文献［CADE11a］）和编辑策略（见本章参考文献［CADE11b］）。我们也实施了数据和来源的信息共享策略，并且允许合作机构基于相互协议来共享信息（见本章参考文献［CADE12］）。

敏捷环境推动策略设计师不断调整或扩展其策略，以快速适应不断变化的环境，从而确保数据集成和组合不会违反数据保密性，特别是在快速操作至关重要（例如情报）时。为了满足这一需求，我们的策略引擎层支持多个策略引擎，而云支持许多策略配置文档。

策略引擎将用户的凭证和可参引 URI 作为输入，然后返回新的 RDF 图（或模型）。他还要在用户接口层之前，对策略的底层逻辑进行评估。可参引 URI 指向一个配置文档，它本身包含其他可参引 URI 指向到机构资源的策略和数据层的机构资源。机构的资源是一个 RDF 文档，三元组处在一个或多个分类级别。例如，一个完整的 RDF 文档如果它包括智能信息，或者含有实际智能信息的三元组子集，它将被分类为敏感类型。因此，一个机构需要超过一种策略来实现对资源细粒度的控制。一个策略因此通过接口定义，它允许每个策略的逻辑实施。在返回一个新的 RDF 图（或模型）到用户接口层之前，策略引擎来评估策略基础逻辑。通过将策略迁移到云中，机构克服了之前可能在策略定义数量上的限制。接下来的小节总结各种策略的类型。在下面的小节里，我们将讨论策略引擎层的细节。这个层包含需要的策略类型，例如，访问控制、编辑、信息共享等。我们将通过依次讨论这些策略类型，反过来激励灵活性策略引擎的需求。

28.2.3.2.1　访问控制策略引擎

访问控制策略授权是一组用户对环境中一组资源执行的一组操作。除非通过一个或多个访问控制策略被授权，否则用户无权访问系统的任何资源。有很多种访问控制策略，可以被分为 3 种主要类型（见本章参考文献［SAMA01］）。这些策略因对用户、操作和对象集（访问控制模型通常将资源看作对象）的限制而不同。这些分类是：①RBAC，它基于角色限制访问；②自主访问控制（Discretionary Access Control，DAC），它是基于用户身份控制访问；③强制性访问控制（Mandatory Access Control，MAC），它基于由授权中心确定的强制性规定控制访问。

基于 RBAC 的策略通常用于简化策略映射的管理，这是 3 类访问控制策略中的一个共同特征。策略创建和可管理性在获取共享资源的访问控制权限方面非常重要。我们使用习惯做法，把权限看作是唯一的（动作、资源）对。给定 n 个资源、m 个用户和一组只有两个动作（读、写），我们最多有 $2 \times n$ 个可能的权限。这给出了 $m \times (2 \times n) = c_1 n$ 个映射。从可能的一组 r 角色中，RBAC 的进一步改进就是分配给两个或多个用户至少有一个角色。因此，我们有 $r \times (2 \times n) = c_2 n$ 个映射，我们也假设 $c_2 \leqslant c_1$。然而，即使有了这个简化，在动态和敏捷社区中实现访问控制水平，所需的策略数量也可能是难以承受的。我们以云为中心的策略框架解决了这一问题，为机构提供支持和扩展其访问控制策略，以满足日益增长的安全需求的能力。

28.2.3.2.2　编辑策略引擎

编辑策略在将文档释放给用户之前，会识别和删除文档中的敏感信息。与限制访问的访问控制策略不同，编辑策略鼓励信息共享，确保在向用户的查询提供最终的 RDF 图（称为修改图）之前，敏感或专有信息被删除（或模糊）。编辑策略依靠转换操作，来规避任何标识或敏感信息。目前支持的策略引擎依赖于图形转换技术，该技术基于图形语法方法（在本章参考文献［EHRI06］、［ROZE97］中提出）。基本上，在有向标记的 RDF 图上应用编辑策略有两个步骤：①识别我们要保护的原始 RDF 图中的资源（或子图），可以使用图查询（即，配有正则表达式的查询）完成；②对此进行修订策略，以图形转换规则的形式识别资源。本章参考文献［CADE11b］中使用此图形转换的实现方法，来修正原始图。

28.2.3.2.3　信息共享策略引擎

信息共享策略允许机构确定其资源共享的背景，或与其他机构的资源相结合。信息共享策略引擎具有用于同时在两个或多个 RDF 图形上，处理查询请求信息的逻辑。我们使用以下 SPARQL 查询来说明这一点。

SELECT $\vec{B}$ FROM NAMED uri1 FROM NAMED uri2 WHERE P

其中 P 是图形模式，$\vec{B}$ 是出现在 P 中变量的元组，uri1 和 uri2 是两个资源 R1 和 R2 的可参引 URI。资源 R1 和 R2 可能来自同一机构，以防机构严格根据保密性问题对其资源进行分区，或者分别属于机构 1 和机构 2 的两个机构。因此，这些资源中的每一个资源都可以定义个人信息共享策略规则。我们定义一个运算符⊙，以便一个信息共享策略通过 uri1 ⊙ uri2 评估。运算符⊙可以通过 RDF 图形实现为图操作。需注意，⊙可以是下面一个运算符：∩、∪、或 -，并且可以被应用于原始 RDF 图，或前一个由运算符⊙得出的图。为了执行运算符⊙，我们以递归方式定义图形。

■ ε 定义为一个图。

■ 图形集合在交、并和集合差运算下是封闭的。G_1和G_2代表两个图，G_1 ［G_2，$G_1 \setminus G_2$和 $G_1 - G_2$是图形，如果 $t \in G_1$ ［G_2那么 $t \in G_1$或 $t \in G_2$；如果 $t \in G_1$ ［G_2，那么 $t \in G_1$且 $t \in G_2$；如果 $t \in G_1 - G_2$，那么 $t \in G_1$且 $t \notin G_2$。

以下 RDF 图表列出了组合策略配置文档的三元组，包含具有共享两个资源的嵌入逻辑策略，R1 和 R2 分别属于两个机构，即机构 1 和机构 2。

```
# entity
<http://policy.org/entity/pol#Combined1_1_1>
  pol:owner <http://policy.org/entity/pol#Agency1>;
    pol:rule <http://policy.org/entity/pol#Cprule1_1_1_1>.

# mappings
<http://policy.org/entity/pol#Cprule1_1_1_1>
  pol:agency<http://policy.org/entity/pol#Agency2>;
    pol:operator "UNION" ;
    pol:type "combined1".
```

这一策略工作于机构级别中。例如，机构 1 将所有资源作为与机构 2 所有资源的联盟共享。策略类型允许机构具有共享模式。类型组合 1 提供了在机构级别上的共享，而

另一种策略类型组合 2，可以更精确地控制机构 1 如何使用机构 2 的资源分类，来确定机构 1 的资源。换句话说，信息共享策略可以包含有关机构的上下文信息，以及资源层面每个资源的元数据。以下显示了我们信息共享策略的两种策略类型：

1）组合 1，$\forall r1 \in Agency1$，$\forall r2 \in Agency2$，用 $r1 \cup r2$。则该策略指出机构 1 将所有资源作为资源联合与机构 2 共享。

2）组合 2，$r1_1$，$r1_2 \cdots$，$r1_n \in Agency1$，用 $r1_1$ ［$r2$，$r1_1 \setminus r2 \forall r2 \in Agency2$。这项策略提供更精细的控制。

28.2.3.2.4　溯源策略引擎

有时，当定义策略时，需要考虑 RDF 图中三元组之间的关系。目前讨论的 3 种策略都没有解决 RDF 图中不同路径敏感信息隐含的情况。本节我们将探讨其他策略引擎。重点是对策略引擎进行定义，以适应访问控制的执行和基于溯源图编辑策略。我们将基于这些策略引擎的逻辑（见本章参考文献［CADE11a］），讨论关于溯源的访问控制策略语言，和如何在溯源上执行编辑（见本章参考文献［CADE12］）。我们将首次给出一个溯源图的例子，以及可能存在于样例溯源图中的来源信息的类型。然后我们将简要介绍在溯源图中执行策略背后的一些理论。

图 28.3 展示了一个情报示例，使用 RDF 表示的溯源图，通过位于一些不友好领域的服务器，概述了文档流（或在另一个机构构成潜在威胁）。这个文档是给记者的。该

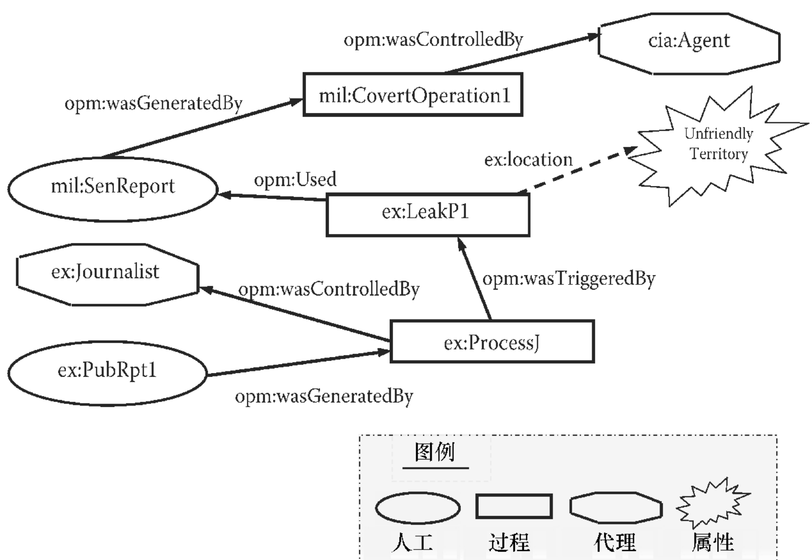


图 28.3　原始图（得到了 Springer 这一科学和商业媒体的允许，内容来自：Cadenhead，T.，Kantarcioglu，M. 和 Khadilkar，V.，Design and Implementation fo a Cloud－Based Assured Information Sharing System，Media from Lecture Notes in Computer Science，Proceedings of Computer Network Security－6th International Conference on Mathematical Methods，Models and Architectures for Computer Network Security，MMM－ACNS 2012，St. Petersburg，Russia，7531，2012，p. 36－50，© Springer，）

溯源图的内容可用于评估文档起源的服务器（即，示例图中的过程）的可信赖性。这个例子溯源图也显示实际来源的基本骨架，其通常只用上下文信息的 RDF 三元组来注释，例如，时间和位置。应注意到，谓词（即弧）用 OPM 抽象谓词（见本章参考文献［MORE10］）标注标签，最终报告可以追溯到 CIA 代理。

图 28.2 中图形嵌入的信息表示一个有向 RDF 图。图 28.3 的原始路径定义如下：

定义 28.1

（原始路径）给定一个来源图，一个原始路径（$s\ p\ o$）是一个路径 $s \xrightarrow{p} o$，基于来源词汇表 V，使用正则表达式定义。

定义 28.2

（正则表达式）令 Σ 为 V 中的术语字母，那么正则表达式集合 RE（Σ）可以归纳定义如下：

- $\forall x \in \Sigma, x \in RE(\Sigma)$;
- $\Sigma \in RE(\Sigma)$;
- $\varepsilon \in RE(\Sigma)$;
- If $A \in RE(\Sigma)$ and $B \in RE(\Sigma)$ then:
- $A|B, A/B, A^*, A^+, A? \in RE(\Sigma)$

符号 | 和符号/分别被解释为逻辑或和组合。

我们的目的是定义两个节点之间的路径，其边缘配备有任意长度的路径，包括长度为 0 或 +，或者至少为长度为 1 的路径的。因此，对于两个节点 x，y，和谓词名 p，x $(\xrightarrow{p})^* y$ 和 x $(\xrightarrow{p})^+ y$；是 G 中的路径。

使用正则表达式扩展的 SPARQL 查询（见本章参考文献［HARR10］），可以定义图 28.3 中原始图的资源（或子图），如下所示：

例 28.1（原始图路径查询）

```
Select ?x
   {ex:PubRpt1 arq:OnPath("([opm:WasGeneratedBy]/
      [opm:WasTriggeredBy]/[ex:location])" ?x).}
```

此查询将返回该位置，并绑定到变量 x，还用于确定原始报告的破解（和泄露）起源。这也可以提醒策略设计者，在其各自的机构中为报告和服务器添加适当的策略。

28.2.3.2.5　策略序列

在一个机构的资源上执行策略会产生一个策略序列。特别地，受保护的资源可以使用多个策略引擎和策略类型的服务。每个策略类型生成其输入 RDF 图的新子图。更要注意的是，策略的效果直接取决于它接收作为输入的 RDF 图。此外，这个策略可能与该策略旨在实现的原始效果不同。序列通过一系列转换获取原始输入图，直到最终的 RDF 图返回给用户。需要注意的是，返回特定 RDF 三元组的策略规则（实现为 SPARQL 查询）的成功，取决于在策略序列中应用规则的转换步骤。我们使用以下 SPARQL

查询来说明这一点：

CONSTRUCT G WHERE P，

G 是一个新构造的图，其中包含一组满足输入图中条件 P 的三元组。保护以下 RDF 三元组的策略，

```
<http://cs.utdallas.edu/semanticweb/Prov-AC/agency#agent_1 >
  foaf:name "John brown";
  foaf:projectHomepage <http://www.agency1.gov/>.
```

will fail if either the name or project home page triple was earlier removed or altered by a previous policy rule.

框架中的策略优先功能有助于机构确定其策略的顺序。在用户接口层中，机构配置策略的顺序。策略序列随后存储在 RDF 序列文件中（使用 RDF 规范“rdf：seq”）。当一个查询被评估时，策略框架将以此按照预定的顺序调用每个策略。

28. 2. 3. 2. 6　规则序列

以类似的方式，可以使用一组规则来实现策略。例如，要完全修改共享资源，机构可能需要单独的规则来修复 RDF 图中每个敏感的三元组。当三元组（或一组三元组）在输入图中符合某些指定标准时，每个规则都会被触发。需要注意的是，每条规则都会转换共享资源的当前状态。因此，规则的每个排序都将影响最终图（也称为修订图）。

28. 2. 3. 3　数据层

数据层是连接工厂，它用作创建连接对象的外观。这些连接对象向策略设计者公开与公共方法相同的属性（功能上）。这使得策略设计人员更容易专注于策略引擎设计。策略设计人员调用 RDF 策略工厂，该工厂返回一个 RDF 模型对象。这个 RDF 模型对象由连接存储支持，可以是一个本地连接、关系数据库连接或云连接。在注册过程中，机构有机会决定其存储资源和配置文件的位置。建议将较小的配置文档存储在本地磁盘（或本地数据库）中，以便快速访问。本地连接消耗较低的带宽，提供实时访问，并在部署之前实现开发。但是，机构可能决定将它们存储在私有云上（或者一个远程数据库服务器上），以利用其中的附加保护。

连接工厂也能使机构将资源存储在任何云基础设施中。例如，机构的资源可以驻留在私有云、社区云或公有云上。私有云部署提供更多的控制，因为机构可以掌管自己的云。社区云提供给特定社区的独占访问，向来都符合合作机构的共同利益。公有云对公众开放，由于对上传到公有云上的数据失去控制，因此容易出现更多的漏洞。机构可以选择混合连接并且同时使用多个部署（例如，混合云模型）。

28. 2. 4　我们策略引擎框架的特点

在下面的小节中，我们将介绍了我们策略引擎框架的一些新特征。

28.2.4.1　策略互惠

策略互惠使机构能够在知道其他机构、资源、策略规范的情况下制定策略。通过注册过程使其成为可能，其中机构可以使用自己的元数据、资源和相关策略。以下讨论提供策略互惠的方案。

如果机构 1 希望与机构 2 分享其资源，那么机构 2 也与其分享资源。但当前的访问控制和编辑策略不提供此互惠性。我们的框架提供信息共享策略，允许机构人员定义基于机构互惠互利的策略。我们在下面提供两个信息共享策略的例子：

1. $\forall r1 \in Agency1, \forall r2 \in Agency2, r1 \cup r2.$

这一策略规定机构 1 分享所有资源与机构 2 的任意资源组成资源的联合（即$\odot \in \{\cup\}$）。

2. $r1_1, r1_2, \ldots, r1_n$□$Agency1$, $r1_1 \cup r2, r1_2 \cap r2, \forall r2 \in Agency2.$

这一策略提供了更精细的控制水平，并定义组合运算符，$\odot \in \{\cap, \cup\}$。

28.2.4.2　条件策略

策略互惠的结果是允许使用有条件的共享策略。例如，机构 1 将其资源与机构 2 分享的前提是机构 2 不与机构 3 分享机构 1 的资源，我们在下面提供一个共享策略的示例信息：

1. $\forall r1 \in Agency1, \forall r2 \in Agency2, Agency1$ **定义了** $r1 \cap r2$. If $\forall r3 \in Agency3$, **那么**
   - $Agency2$ **不能定义形如** $r1 \cap r3$ **的共享策略，**
   - $Agency2$ **不能定义任何形如** $r1 \subseteq r2 \odot r3$ **的共享策略，**
     当$\odot \in \{\cup, \cap\}$**时**

28.2.4.3　策略对称性

策略互惠的另一个结果是在分享策略时具有对称性。例如，机构 1 与机构 2 用操作符$\odot$共享资源。假设机构 2 也分享资源与机构 1 使用同样的操作符$\odot$。我们在下面提供信息共享策略的示例：

1. $\forall r1 \in Agency1, \forall r2 \in Agency2, Agency1, r1 \cup r2$ **如果** $Agency2$ also uses $r2 \cup r1$.

28.2.4.4　制定和衡量策略

为了在信息环境中实现机动性的自由，并提供信息的力量来确保任务的完成，一个机构应该有能力根据情况快速制定策略并进行部署。接下来，我们将讨论在制定策略期间和之后可以使用的功能。

28.2.4.4.1　策略制定

机构 1 希望模拟现实环境并创建测试场景，以便可视化每个策略配置的结果。我们的策略框架提供 3 种配置：①用于开发和测试的独立版本；②关系数据库支持的版本；③实现高可用性和扩展性的基于云计算的版本，同时保持低的配置和操作成本。

28.2.4.4.2　排序效果

机构 1 希望根据用户凭证，将结果集返回到用户的查询。策略序列特性可以通过排列策略以及其各自的规则，来配置不同的结果。

28.2.4.4.3　高速弹性

机构 1 能识别其现有策略配置中最新的安全漏洞，并希望以更精细的粒度来支持策略以及扩展（或扩大其）现有策略。我们的策略引擎提供了所有策略都要实施的策略接口。因此，我们可以根据需要添加更新的策略。另外，我们的策略引擎赋予机构快速的弹性，因此我们的策略框架可用的功能看起来是无限的。

28.2.4.4.4　位置独立性

机构 1 希望其资源存储位置更靠近其使用的地方，但策略层几乎不做改变。我们的策略引擎提供位置独立性，即使策略引擎对资源的确切位置没有控制或未知，也可以使用连接管理器通过指定位置来访问资源。注意一个机构的资源可以在任何云、任何地理位置。通过可参引 URI，对定位任何资源的能力提供了很大的灵活性。

28.2.4.4.5　部署模式

机构 1 可以利用不同的部署模式。比如，私有云、混合云、社区云或公有云。连接管理器允许机构根据不同的风险因素和目标，在连接类型列表中进行选择，以保护数据的机密性。

28.2.4.5　合理利用资源

起源提供了关于信息被操纵的原因以及对信息操纵来源追踪的解释。这确立了机构之间的信任，从而促进了共同目标的伙伴关系。

机构 1 要求机构 2 合理利用资源 R2。目前商业访问控制策略主要用于保护单个数据项目，而目前的编辑策略是为了修改文本和图像而设计的。我们的策略引擎允许机构人员定义策略来源。因此，机构 2 可以提供溯源信息给机构 1，但是使用访问控制和编辑策略保护它。

28.2.4.6　策略规范和执行

我们的架构设计支持高级策略规范，从而将业务层与特定的策略分开。

机构 1 希望用高级语言（例如 XACML）来表达策略，并且不选择学习 RDF 或其任何变体。该框架提供用户和策略引擎层之间的 Web 接口层，从而使用户可以独立于策略的实际情况来指定其策略。一个合适的适配器，也成为数据转换器，将把每个高级策略规范转换成为用适当的 RDF 表示的适当策略，从而保护机构的资源。

可以使用比 RDF 更具有表达能力的语言来指定策略，比如通过使用形式词汇扩展 RDF，特别是 OWL 子语言。OWL 具有基于描述逻辑的形式语义，这是一阶逻辑的可判断片段。因此，通过支持这种适配器模式，我们的框架被扩展，以处理 OWL 中指定的语义策略，高级策略可以使用现有或定制的转换器，转换成适合的 OWL 子语言。

28.3　总结和展望

本章介绍了在云中运行的第一个 AIS 框架的设计和实现。如前所述，这个想法是每

个组织将其数据和信息共享策略存储在云中，根据策略共享信息。我们的框架由一个以信息共享为目的的策略引擎，以及一个在云中存储和查询数据的安全数据引擎组成。我们还用示例策略描述了我们系统的运行。

我们的框架是灵活的，可以添加额外的数据源和云。此外，通过使用 RDF 作为策略引擎，我们可以添加更复杂的信息共享策略。这是我们系统的主要优势之一。未来的发展方向包括制定和推理更复杂的策略，以及在现实环境中测试我们的系统。

参 考 文 献

[BRAU08] Braun, U., A. Shinnar, and M. Seltzer, Securing provenance. In *Proceedings of the 3rd Conference on Hot Topics in Security*, USENIX Association, San Jose, CA, 2008, Article No. 4.

[CADE11a] Cadenhead, T., Khadilkar, V., Kantarcioglu, M. and B. Thuraisingham, A language for provenance access control. In *Proceedings of the first ACM Conference on Data and Application Security and Privacy*, San Antonio, TX, pp. 133–144, 2011.

[CADE11b] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. M. Thuraisingham, Transforming provenance using redaction. In *Proceedings of the 16th ACM Symposium on Access Control Models and Technologies*, Innsbruck, Austria, pp. 93–102, 2011.

[CADE12] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. Thuraisingham, A cloud-based RDF policy engine for assured information sharing. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*, Newark, NJ, pp. 113–116, 2012.

[CADE12] Cadenhead, T., Kantarcioglu, M., and Khadilkar, V., Design and Implementation of a Cloud-Based Assured Information Sharing System, Media from Lecture Notes in Computer Science, *Proceedings of Computer Network Security-6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012*, St. Petersburg, Russia, 7531, 2012, p. 36–50. © Springer.

[EHRI06] Ehrig, H., *Fundamentals of Algebraic Graph Transformation*. Springer-Verlag, New York Inc, 2006.

[HARR10] Harris, S. and A. Seaborne, SPARQL 1.1 Query Language. *W3C Working Draft,* 2010, www.w3.org/TR/spargl11-query/

[KAGA02] Kagal, L. REI., A policy language for the project, http://www.hpl.hp.com/techreports/2002/HPL-2002-270.pdf. 2002.

[KLYN04] Klyne, G., J. J. Carroll, and B. McBride, Resource description framework (RDF): Concepts and abstract syntax. *W3C Recommendation*, 10, 2004, www.w3.org/TR/rdf-concepts/

[MORE10] Moreau, L., Clifford, B., Freire, J. et al. The open provenance model core specification (v1. 1). *Future Generation Computer Systems,* 27(6), 743–756, 2011.

[PRUD06] Prud'hommeaux, E., A. Seaborne, (Eds.). SPARQL query language for RDF. *W3C Working Draft*, 4, 2006, www.w3.org/TR/rdf-spargl-query/

[ROZE97] Rozenberg, G. and H. Ehrig, *Handbook of Graph Grammars and Computing by Graph Transformation*. World Scientific, River Edge, NJ, 1997.

[SAMA01] Samarati, P. and Vimercati, S. de. Access control: Policies, models, and mechanisms. *Foundations of Security Analysis and Design*, 137–196, 2001.

第七部分总结

本部分描述了我们为实现安全应用程序而开发的云实验系统。这些应用程序包括恶意软件检测、内部威胁检测和信息共享保障。

在第 25 章中，我们介绍了 EMPC，一种用于无限长度、概念漂流的自动分类的新型集成学习技术。我们已经说明，通过将 EMPC 应用于从多态恶意软件中获取的真实数据流和僵尸网络流媒体样本，比其他流数据分类技术要拥有更好的检测精度。我们算法的实现是在云上进行的。

在第 26 章中，我们讨论了内部威胁检测的方法。我们将内部威胁和它们之间的通信作为 RDF 图，然后通过查询和挖掘图来抽取信息。另外，我们也提供了一个基于云的内部威胁检测的综合框架。

在第 27 章，描述了我们开发的基于云端信息共享安全系统的 CAISS 方法。在实现 CAISS 时利用了我们开发的几种技术以及开源工具。我们也描述了一种理想的基于云计算的信息共享保障系统的设计，它叫作 CAISS + + 。

在第 28 章中，我们描述了运行在语义云中的第一个信息共享保障框架的设计与实现。语义云是向消费者提供语义 Web 服务的云。这个想法是为每个组织将其数据和信息共享策略存储在云中。信息根据策略共享。我们描述了一种利用语义 Web 技术的基于云的信息共享框架。该框架由一个以信息共享为目的的策略引擎，以及一个在云中存储和查询数据的安全数据引擎组成。我们也用策略的示例描述了系统的运行情况。

第八部分　迈向可信赖的云

第八部分简介

前面大部分章节的讨论都集中在云的安全性上。需要注意的是，虽然安全性通常包括机密性、完整性和信任，但是我们对安全性关注的重点只是机密性。在第八部分，我们将讨论开发可信赖云的其他方面，包括信任、隐私和完整性。

第八部分由三章组成：第 29 章、第 30 章和第 31 章。第 29 章讨论信任管理和云服务。信任基本上是关于你对于一个人说什么，或者对那个人可以保留多少信心。信任也可以衡量一个人是否会履行自己的承诺。一般来说，在向某人发出信息之前，即使他有权从我这里获取这些信息，我也要确定他是否可以信任。第 30 章专注于隐私和云服务。需要注意的是，目前已经提出了关于隐私的几种不同的定义。我们在这里使用的定义是，一个人有权决定要公开什么信息。因此，任何组织违反本人意愿就违反了本人的隐私。第 31 章重点介绍完整性和云服务。我们的完整性包括数据的准确性，以及数据质量和数据来源。

第 29 章　信任管理和云计算

29.1　概述

本章重点介绍云服务的信任管理。在开发的安全系统中已经对信任进行了大量的讨论。早期大部分的重点是用可信软件来开发高度保障的系统。例如，在设计时，要根据可信计算机系统评估标准（Trusted Computer Systems Evaluation Criteria，TCSEC），按 A1 级进行评估多级系统时，该软件必须经过正式的验证过程，以确保没有隐蔽的通道。这样的软件被称为可信软件。然而，随着数据和应用程序的安全问题不断增多，这在过去十年中变得尤为突出，重点是信任个人或代表个人行事的过程。在这里，我们一定要确定必须放在个人身上的信任。此外，数据也必须被分配信任值。也就是说，如果数据从可靠的个人或来源（例如文件或数据库）发出，那么数据可能就具有高信任值。

云服务也需要有可信度从而进行某些操作。执行命令和控制以及患者监控等关键功能的云服务，比其他像查找销售鞋类公司的云服务要更加具有可信赖性。在本章中，我们将讨论与信任管理相关的问题，然后讨论基于信任的云服务。我们还将讨论对语义云服务的信任。需要注意的是，这种云服务将语义 Web 技术用于如数据、策略表示和推理等任务。

本章的结构如下。29.2 节将讨论信任管理，包括信任个人和数据。特别是在 29.2.1 节对信任管理和信任协商进行讨论。需要注意的是，信任和风险之间有一定的关系。也就是说，如果一个人是不值得信任的，而你又必须给他/她一些数据，那么你就正在冒险。因此，29.2.2 节讨论相关信任和风险的一些发展。29.2.3 节讨论基于声誉的信任。然后在 29.3 节，我们将讨论信任管理和云服务。尤其是作为云服务的信任管理将在 29.3.1 节中讨论。29.3.2 节将讨论对云服务的信任。在 29.3 节对本章进行总结。图 29.1 说明了信任的各个方面。

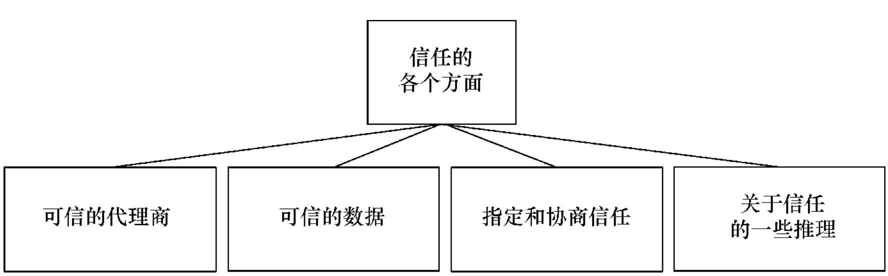


图 29.1　信任的各个方面

29.2 信任管理

29.2.1 信任管理与协商

在讨论信任管理的各个方面，以及描述与语义云服务的关系之前，我们需要确定什么是信任。信任是由哲学家定义的，它涉及一个人对另一个人寄予多少价值。这个值取决于这个人是否可以保密或进行安全的活动等。在基于对某人信任的基础上，从某人发出的数据也将被分配一个信任值。稍后我们会处理数据信任。首先，我们将重点关注个人的信任。我们可以扩大论点，不仅包括个人，还包括一群人，甚至一个网站或组织。

信任工作最初着重于为确保软件符合规范，而必须进行的验证或测试的数量。如果软件有一个特洛伊木马，那么它就是不值得信任的。如果软件是可信赖的，则根据用于信任软件的技术（例如，形式验证与测试），就可以确保软件是安全的。后来，随着数据安全越突出，信任被分配给个人或组织。在这种情况下，使用两种方法来定义信任：一个是基于凭证，另一个是基于声誉。在信任的研究界，这两种学派都受到重视。

Bertino 及其团队对基于凭证的信任管理进行了广泛的研究。研究的想法是交换个人之间的凭证，根据凭证的类型，双方之间建立信任。凭证最初是通过一些凭证授权机构获得的。例如，如果约翰想要看到简的个人资料，他必须向简出示他的凭证，该凭证由一个凭证机构发给他。另一项研究，着重以声誉为基础的信任管理，是 Winslett 及其同事以及 Winsborough 等人的工作。ACM、SACMAT 和 IEEE 策略等会议记录中出现了许多基于凭证信任管理的论文（另见本章参考文献［BERT03］、［YU03］、［WINS04］）。

在基于声誉的系统中，信任是根据他以前的行为获得的声誉来分配的。例如，如果简适合的职位是老师，那么听到过简的人就将会讨论她的声誉，如她不可靠，并且逃了大量的课程。如果是这样，那么简作为老师的声誉并不好。所以，简不会被信任而得到这份工作。我们在日常生活中一直使用声誉。也就是说，我们基于其声誉来信任个人或组织。通常很难去提高声誉。然而，这并不会损害声誉，也不会造成信任价值的降低。基于声誉的信任系统在本章参考文献［SHMA］中讨论。第三种类型的信任是确定数据的置信度值。换句话说，你对数据抱有多大程度的信任？为了给出答案，我们需要确定谁创造了数据？谁访问过数据？数据是否经过不可信的组织？在第 31 章讨论数据质量和数据来源时，我们将讨论数据信任。

一旦信任值被分配，管理信任需要什么？这涉及根据信任值进行交换数据，以及根据所接收或丢弃的凭证，或改变的声誉，来增加和减少信任值。例如，如果约翰被委任处理一些关键数据，然后知道约翰滥用了这些数据，那么其信任值就会降低。现在，出现了信任形式化和信任行为规范化的研究。用于信任管理的代数也正在开发中。信任管理的一个重要方面是信任协商。在这里，双方可以相互协商它们的信任值和它们之间要共享的数据。信任协商是信任管理的重要研究领域（见本章参考文献［WINS04］）。信任协商过程如图 29.2 所示。

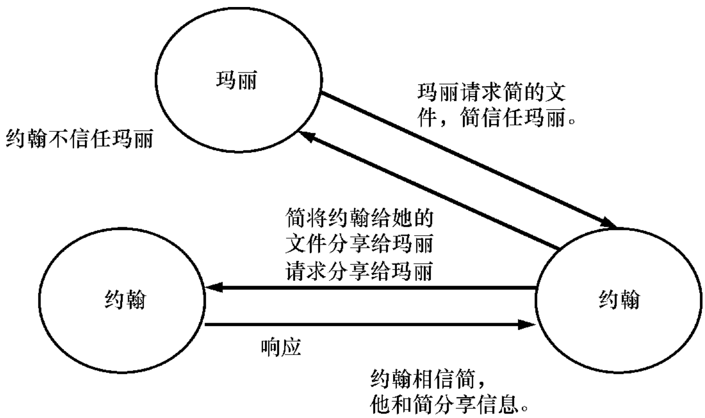


图 29. 2　信任协商

29. 2. 2　信任与风险管理

为了管理数据共享的风险，我们需要了解风险因素。虽然信任和风险相关，但它们有区别。例如，你对某人越信任，你与此人分享的数据越多。然而，也有 A 医院信任 B 医院的情况，但是 A 不与 B 共享数据，因为 B 的系统不安全。人们也可以这样认为，由于 B 的计算机不安全，所以 B 不能被信任。在某些情况下，与不可信任方共享数据可能也不会有风险。例如，即使医院不信任药品公司，医院也可以与药品公司分享其数据，以寻求治愈方法。换言之，有人可能会认为，即使药品公司不能适当地使用数据，医院也会一定程度上相信能找到治愈疾病方法的药品公司。但是，如果数据不敏感，那么分享这些数据可能就不是一个问题。因此，人们可以把信任和风险相互关联，但是这两者是不同的概念。

虽然提出了信任与风险之间关系的不同模式，但数据共享应用中的信任与风险之间的确切关系尚未明确。我们需要的是制定信任和风险关系的适当模型。信任不是影响风险的唯一因素。我们的研究涉及理解信任和风险，并开发基于信任的风险模型。为了创建基于信任的风险模型，我们需要捕获与信任错误判断相关的所有风险。此外，他指出，即使风险很高，也要进行成本效益分析来确定是否要分享数据。信任风险管理如图 29. 3 所示。

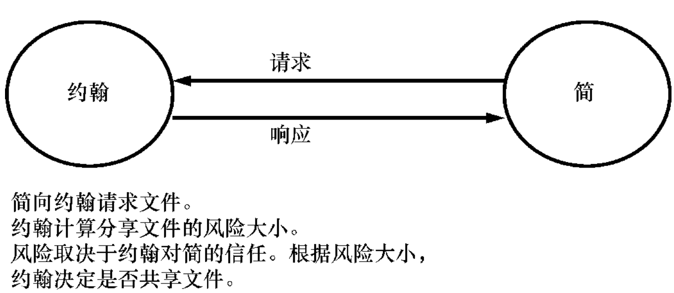


图 29. 3　信任和风险

29.2.3 基于声誉的系统

可以使用所谓的声誉网络建立信任。如本章参考文献［GOLB03］所述，声誉的网络是一个分布式的基于 Web 的社交网络。信用评级从一个用户推断到另一个用户。个人连接到他们评价的每个人，并导致大量用户的网络互联。这里唯一的要求是个人应该在网络中相互表明他们的声誉评级。个人控制自己的数据，数据以分布式方式进行维护。数据可以存储在任何地方，并通过共同的基础进行集成。

FOAF（见本章参考文献［RDF］）项目说明了语义 Web 和声誉网络之间的关系。本体论词汇用于描述人们及其关系。这通过提供描述声誉关系的机制得到延伸，并允许人们评价另一个人的声望或可信度。

我们正在开发算法以推断声誉。如本章参考文献［GOLB03］所述，向一个人（来源）提供关于另一个人的声誉的建议。信任和声誉相关文献包含许多不同的研究指标。这些指标根据用于进行计算的角度而进行分类。例如，全局度量为网络中的每个实体计算单个值，而本地度量计算网络中个人的声誉评级。在全局系统中，实体将始终具有相同的推断等级。但在本地系统中，可以根据推断的节点对实体进行评估。

声誉系统的一个例子是 TrustMail。它是一个消息评分系统，并为消息的文件夹视图添加了声誉评级。在用户看到声誉评级后，它有助于相应地对消息进行排序，突出了重要和相关的信息。图 29.4 所示为声誉网络。

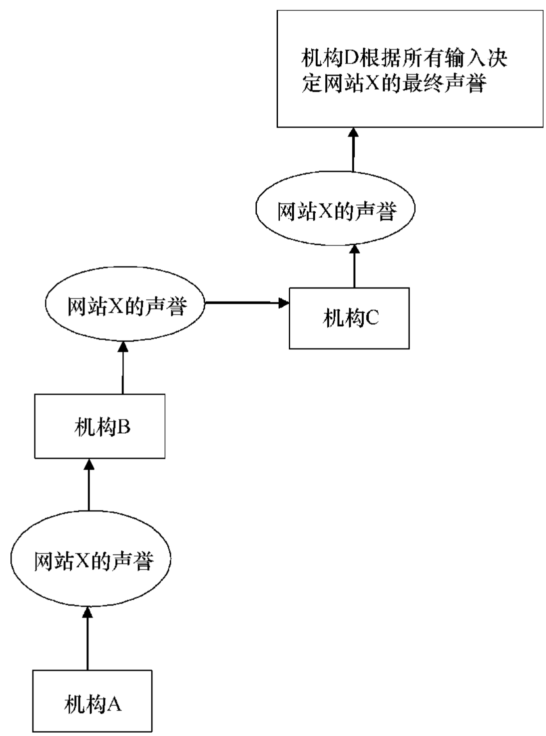


图 29.4 声誉网络

29.3 信任和云服务

29.3.1 信任管理即云服务

这里有两个方面的含义。一个是作为云服务或云服务的集合实现信任管理。另一个是探索云服务的信任管理。我们勾画了将信任管理作为云服务（或 Web 服务）实现的情况，如图 29.5 所示。用户向 Web 服务器或 CSP 发出请求以获取资源。查询服务将向

云提供的信任管理服务发出请求，以确定用户的信任级别。在此信任级别的基础上，向用户提供适当的资源。

我们也有兴趣使用语义 Web 技术进行信任管理和协商。虽然已经开发了几种信任策略语言，但是此处介绍利用 XML 作为策略所表示的知名系统，这是由 Bertino 及其小组在米兰大学和普渡大学开发的系统。开发的系统被称为 Trust - X，是基于 XML 的。基于凭证的系统 Trust - X 使用基于 XML 的信任策略语言（见本章参考文献［BERT04］）。回想一下，利用语义 Web 技术来表示和推理策略的云服务，就是语义云服务。

虽然 XML 是一种合适的策略语言，但它的缺点在于它不能充分表示语义。例如，当且仅当 B 不信任 C 时，A 信任 B，或者有 A 信任 B 且 B 信任 C，这样的语句并不意味着 A 信任 C。在 XML 中很难表示这些语句。需要注意的是，与 XML 不同，RDF 可以表示类 - 子类关系，OWL 等语言也可以表示如联合和交集的关系。因此，我们需要丰富的策略语言来表示信任。此外，自 9/11 委员会报告（http：// www.9 - 11commission. gov/report/911Report. pdf）以来，环境正在从需要获知转移到需要分享。因此，在这样的环境中表示信任关系是非常重要的。我们需要策略语言来表示“紧急情况”下的声明形式，需要共享所有数据，然后确定信任方面的数据共享所产生的后果。“Finin 及其同事正在研究使用 REI 等语言进行“需要分享”的环境（见本章参考文献［KAGA03］）。

使用基于语义 Web 的策略语言的优点，是可以使用基于描述逻辑的推理能力来推理信任语句，并对未明确指定的信任进行推论。人们也正在探索 Jena 和 Pellet 等推理引擎，用于表示和推理基于语义网络的策略规范。在麻省理工学院（Massachusetts Institute of Technology，MIT）进行的策略意识网络项目，也在开发信任策略的规范语言和推理引擎。

需要注意的是，语义 Web 层中包括逻辑、证明和信任。这种类型的信任与本章所讨论的信任不同。语义 Web 的信任层本质上是关于语句可信度的推理。例如，你对“约翰和詹姆斯是最好的朋友”这些声明的信任度。相信这个声明取决于声明的来源。当我们在后面的章节中讨论数据质量和来源时，我们将讨论此类型的信任。

虽然现在有很多关于策略语言规范的研究，但是网络语言的优势在于我们可以利用正在开发的推理工具来推理策略，以便我们检查策略的一致性。我们也希望确保信任策略不泄露分类或私有的敏感信息。Bertino 和她的工作组（见本章参考文献［SQUI06］）已经进行了这些研究。图 29. 5 说明了用于信任管理的云服务，而图 29. 6 为用于信任管理的语义云服务。



图 29. 5　用于信任管理的云服务

29.3.2　云服务的信任管理

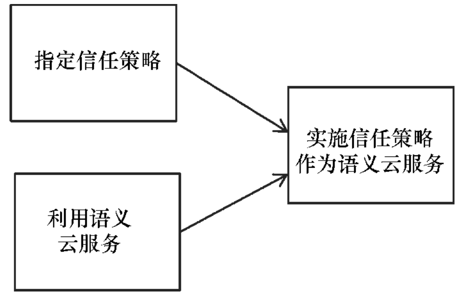


图 29.6　信任管理的语义云服务

在前面的小节中，我们讨论了云服务和语义云服务在信任管理中的应用。基本上，这里的想法是将信任管理实现为云服务，使用如 XML、RDF 和 OWL 之类的语言来指定策略，以及基于描述逻辑策略的原理和调用云服务来管理信任。在本节中，我们将讨论如何将信任管理技术应用于云服务以及语义云服务。需要注意到，语义 Web 是一种技术集合，可以为我们提供机器可理解的网页。因此，这里面临的挑战是我们如何相信为获得机器可理解的网页而进行的推理。此外，我们会相信产生的网页吗？关于云服务的信任管理，这样的想法是决定我们对云服务的信任度。云服务是否经过验证？如果是，认证级别是多少？图 29.7 说明了对云服务的信任，而图 29.8 则说明了语义云服务的信任。



图 29.7　云服务的信任管理



图 29.8　语义云服务的信任管理

语义云服务的层次之一是逻辑、证明和信任层。在这里，我们需要技术来推理网页的准确性。我们相信产生的数据吗？我们是否信任代表用户的代理进行活动所做出的决策？另外，信任网页还将决定谁能生成网页。如果生成网页的代理非常值得信赖，那么我们可能会对结果更加信任。我们将在第 30 章中讨论数据质量和数据来源的这个方面。

另一方面是关于信任代理，作为 Web /云服务的集合被实现，其利用基于 XML 和 RDF 的语义 Web 技术来执行活动。我们能相信代理人的答案吗？这些代理是否在他们之间进行信任协商？也就是说，代理之间的信任基本上是人与人之间建立的信任。该信任可能取决于凭证或可能基于声誉。例如，在提供旅游服务时，代理必须预约，预订酒店，并安排客户参与旅游。代表客户行事的代理将以 XML 或 RDF 阅读网页，然后联系代理航空公司和酒店代理。第一个代理商对另一个代理商的信任可能取决于旅行社拥有的凭证或声誉。

因此，当我们讨论信任时，有两个主要方面。一个是对数据的信任，另一个是对代理的信任。对数据的信任将取决于对代理的信任。类似地，代理持续销售值得信赖的数据可以被认为具有较高的信任值。

29.4 总结和展望

在本章中，我们讨论了信任管理，及其与云服务和语义云服务的关联。我们首先讨论了信任管理的各个方面，包括定义信任和描述信任协商。然后，我们又讨论了在语义云服务的环境中实施信任。此外，我们还讨论了使用语义云服务技术来指定信任策略。最后，讨论了相关概念，包括基于风险的信任管理和声誉网络。

我们的目标是提供一个高级视图，能够说明在信任方面与 Web 服务相关的挑战是什么，进展如何。信任管理是一个新兴的研究领域，包括普渡的 Bertino、麻省理工学院的 Berners Lee、UMBC 的 Finin（马里兰大学，巴尔的摩县）和 Winslett 在 UIUC（伊利诺伊大学，厄本那香槟分校）等，对此进行了广泛的研究。例如，UMBC 的 Finin 和同事们开创了使用称为 REI 的语言，来指定和推理信任的技术。我们正在与 UMBC 合作，在需要共享的环境中进行信任管理。虽然已经提出了许多信任协商方法，但我们仍需要研究评估这些方法，并确定在什么情况下哪些方法是适当的。因此，尽管在过去十年中我们对信任管理做了很多工作，但对于包括云服务在内的特定应用和领域，仍然有许多工作要做。

参考文献

[BERT03] E. Bertino, E. Ferrari, and A. C. Squicciarini, Trust-Chi: An XML framework for trust negotiations. *Communications and Multimedia Security*: 146–157, 2003, Torino, Italy.

[BERT04] E. Bertino, E. Ferrari, and A. C. Squicciarini, Trust-X: A peer-to-peer framework for trust establishment. *IEEE Transactions on Knowledge of Data Engineering* 16(7): 827–842, 2004.

[GARC06] R. García and R. Gil, An OWL copyright ontology for semantic digital rights management, *Workshop on Ontology Content and Evaluation in Enterprise*, 2, 2006, Montpellier, France.

[GOLB03] G. Jennifer, B. Parsia, and J. Hendler, Trust networks on the semantic cloud service, *Proceedings of Cooperative Information Agents*, August 27–29, 2003, Helsinki, Finland.

[KAGA03] L. Kagal, T. W. Finin, and A. Joshi, A policy based approach to security for the semantic web. *International Semantic Web Conference* 2003: 402–418, White Maris, NY.

[RDF] RDFWeb: FOAF: The Friend Of A Friend vocabulary. http://xmlns.com/foaf/0.1/.

[SHMA06] V. Shmatikov and C. Talcott, Reputation-based trust management. http://www.cs.utexas.edu/~shmat/shmat_rtm.pdf

[SQUI06] A. C. Squicciarini, E. Bertino, E. Ferrari, and I. Ray, Achieving privacy in trust negotiations with an ontology-based approach. *IEEE Transactions on Dependable Security Computing* 3(1): 13–30, 2006.

[WINS04] W. H. Winsborough and N. Li, Safety in automated trust negotiation. *IEEE Symposium on Security and Privacy* 2004, Berkeley, CA.

[YU03] T. Yu and M. Winslett, A unified scheme for resource protection in automated trust negotiation. *IEEE Symposium on Security and Privacy* 2003: 129–132, Berkeley, CA.

第 30 章　隐私和云服务

30.1　概述

保密性是云仅向基于策略被授权的人员发布数据/信息，而隐私是关于一个人确定他应该发布什么信息。因此，如果 CSP 的隐私策略不能被该用户接受，则他/她可以决定他/她是否想将信息存储在云中。

但是，需要注意的是，即使隐私权在国会级别上已经被大量讨论，但并不是每个人都同意这一定义。例如，作者在华盛顿特区武装部队通信和电子协会的非分类级别上，讲授数据挖掘、国家安全和隐私。参加课程的学生主要是为国防和情报机构工作。对于他们来说，隐私并不是说出他或她病历的那种感觉。据我所理解，联邦调查局（Federal Bureau of Investigation，FBI）的隐私理念是确保美国公民的个人信息不会陷入不法之人手中。即使是其他机构，只有当该机构有权获取该信息时，联邦调查局才会发布私有信息。在某种程度上，隐私或多或少就像这些相关组织的机密性。

人们已经开展了许多关于隐私的工作，包括隐私策略的规范和执行、开发隐私保护技术、数据挖掘和隐私标准。W3C 的一个挑战是规范网站（或 CSP）可以用来指定其称为 P3P 的隐私策略的标准。另一个挑战是确保私有信息不会由于语义 Web 挖掘（云中的数据挖掘）而被公开。最后，云服务必须确保私人信息不泄露。同样，进行云挖掘的云服务必须确保个人的隐私得到保护。

本章的结构如下。在 30.2 节，我们讨论一般隐私管理。特别是，30.2.1 节讨论隐私问题。30.2.2 节讨论隐私约束处理和数据挖掘的隐私问题。P3P 将在 30.2.3 节中讨论。30.3 节讨论隐私和云服务之间的关系。特别地，隐私作为云服务将在 30.3.1 节中讨论。30.3.2 节讨论云服务的隐私。本章结束于 30.4 节。

30.2　隐私管理

30.2.1　隐私问题

社会科学家多年来一直在研究隐私，策略专家也为机构和公司制定了隐私策略。但是，最近安全专家才开始关注隐私。此外，针对 DARPA 的恐怖主义信息意识计划，以及对数据挖掘的关注，这已经促进了在隐私保护数据挖掘和隐私保护数据管理方面取得进展。今天，隐私是信息安全的重要领域。然而，由于每个组织和机构都有不同的看法，所以很难对隐私进行精确定义。

那么问题是：什么是隐私？一般的观点是，一个人决定他或她应该发布什么样的个人信息。在我们拥有数据分析、数据挖掘和 WWW 的工具之前，这样的定义是正确的。通过这些工具，现在可能有人推断关于另一个人的私人信息。因此，我们需要重新定义隐私的概念。另一方面，一些组织想要控制关于社区的个人信息，并决定谁应该发布个人信息。也就是说，作者的理解是联邦调查局有关于个人的信息，他们将有权决定是否将信息发布给中情局。最初，作者认为这基本上是保密性而不是隐私。然而，在更多地关注隐私问题和阅读这个主题之后，作者现在认为，隐私的定义并不是被普遍定义的那样。隐私必须由组织定义。也就是说，一个组织可以将隐私策略定义为保护其敏感信息的策略。另一个组织可以将隐私策略定义为，在组织内工作的人员指定哪些信息可以由他们发布。因此，是否隐私策略是保密策略的一部分，或者是单独的隐私策略，要留给组织来确定。

我们的兴趣还在于隐私、机密性和信任之间的关系。正如我们前面的章节所讨论的，在我们的工作中，我们做出了以下假设。在 CSP 和基于凭证或声誉的用户之间建立信任。当用户登录云进行交易时，CSP 将指定其隐私策略。然后用户将确定他是否要输入个人信息。也就是说，如果 CSP 将向第三方发出用户的地址，则用户可以决定是否输入该信息。然而，在用户输入信息之前，用户必须决定他是否信任 CSP。这可能要基于凭证和声誉。如果用户信任 CSP 而且他对该策略感到满意，则用户就可以输入他的私人信息。如果不信任 CSP，那么他可以选择不输入信息。

我们给出了类似的对机密性的推理。这里，用户正在从 CSP 请求信息，CSP 检查其机密性策略，并决定向用户发布哪些信息。CSP 还可以检查其对用户的信任，并决定是否向用户提供信息。还可以基于对用户或 CSP 的信任来确定数据的质量。

有关隐私和语义云服务的更多具体细节方面将在接下来的几节中讨论。具体来说，包括应用语义 Web 技术进行隐私管理、语义云服务的隐私问题、P3P、通过推理发生的隐私问题以及隐私保护云挖掘等。图 30.1 说明了隐私管理的各个方面。

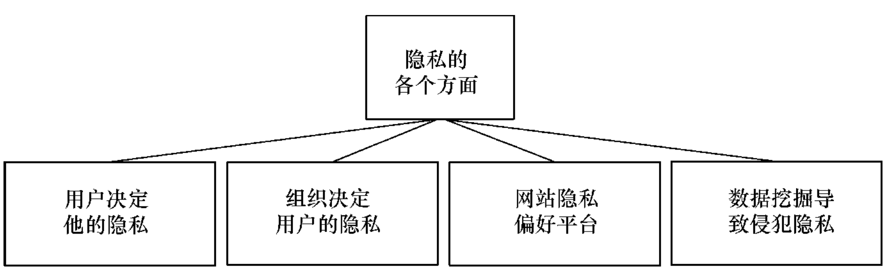


图 30.1　隐私管理

30.2.2　推理产生的隐私问题

我们对安全数据库的推理问题进行了广泛的研究。大部分工作侧重于安全约束处理，现在它已经被称为策略管理。这些策略包括内容和上下文相关约束，以及基

于动态和事件的约束。例如，在战争开始之后，该船的使命就成为机密（见本章参考文献［THUR90］）。因此，我们采用这种隐私约束处理方法，其中的安全级别现在将成为隐私级别（公共、私人、半公共等），安全约束将成为隐私约束，如名称和保健记录在一起。应该指出的是，通过这种方法，我们假设隐私和保密是一样的。现在，这也符合 FBI 关于隐私的概念，它必须保护美国公民的私人信息。但这并不符合医疗隐私，在这种情况下，隐私由个人指定。也就是说，个人决定了他必须保密的信息。隐私控制器由个人管理。客户将决定如果他/她发出他/她的遗传信息，那么一家保险公司就可以找出他/她可能容易出现的疾病。因此，隐私控制器将引导客户了解有关他/她的哪些信息。

在这里，使用如 XML、RDF 之类的语义 Web 技术所表示的数据，用推理引擎来增强本体。这些引擎可以执行规则处理或利用基于本体的推理，从现有数据中推导出新的数据。如果新数据是私有的，那么他们可以向客户提供关于什么信息应该保密的建议。需要注意的是，在联邦调查局，隐私控制器本质上是机密性控制器（我们称之为推理控制器），因此它作用于服务器端，并确定向客户端（如 CIA）发布哪些信息。

值得注意的是，我们已经证明推理问题是无法解决的（见本章参考文献［THUR90］），也已经应用了类似的技术来证明隐私问题是无法解决的（见本章参考文献［THUR06b］）。图 30.2 为隐私控制器的架构。

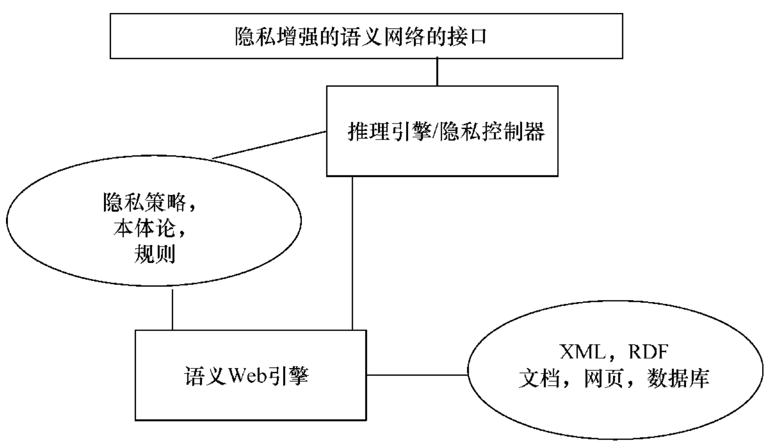


图 30.2　隐私控制器架构

30.2.3　隐私偏好平台

P3P 是一种新兴的行业标准，使 CSP 能够以标准格式表达其隐私惯例。策略的格式可以由用户代理自动检索和理解。它是 W3C（www. w3c. org）的产品。正如我们所说，在许多领域考虑的隐私和安全性的主要区别如下：①向用户通报 CSP 执行的隐私策略；

②用户一般不知道安全（或保密）策略；③当用户进入云端时，CSP 的隐私策略将被传送给用户；④如果隐私策略与用户偏好不同，则就会通知用户。然后，用户可以决定如何继续。

几家大型企业正在开展 P3P 标准，包括 Microsoft、IBM、HP、NEC、诺基亚和 NCR。几个网站也实施了 P3P。语义 Web 工作组已经采用 P3P。P3P 的初始版本使用 RDF 指定策略，最近的版本已经迁移到 XML。P3P 策略使用 XML 和命名空间进行编码策略。

示例：目录购物。您的姓名不会发给第三方，但您的购买内容将会发给第三方。

```
<POLICIES xmlns=http://www.w3.org/2002/01/P3Pv1>
    <POLICY name=- - - -
    </POLICY>
    </POLICIES>
```

P3P 有自己的声明和 XML 表达的数据类型。P3P 模式使用 XML 格式。XML 是理解 P3P 的前提条件。对 2005 年 1 月发布的 P3P 规范，这里使用目录购物的例子来解释其概念。P3P 是一个国际标准，也是一个正在进展的项目。

需要注意的是，P3P 代表不了法律，但 P3P 与法律一起工作。如果网站不遵守他们的 P3P 策略，就必须采取适当的法律行动。今天，XML 是指定 P3P 策略的技术。策略专家将必须明确策略，技术必须制定规范。如果违反策略，法律专家将不得不采取行动。

30.2.4 保护隐私的云挖掘

在上一章（见本章参考文献［THUR05］）中，我们讨论了基于隐私保护的数据挖掘。这个想法如下。目前，使用数据挖掘工具，即使没有经验的用户也可以得到未经授权的推论，这些推论可能是非常敏感或私有的。进而，我们的目标就是隐藏私人数据，例如某个人的疾病，同时给出一般的趋势和关联。也就是说，我们可以放出“生活在加利福尼亚的人更容易患哮喘”的信息，而不会发出约翰有哮喘的事实。基于隐私保护的数据挖掘技术，往往与干扰数据或随机数据一起工作，而不会泄露实际数据。

最近有关于语义云挖掘的报道。在此提及两个方面。一方面是挖掘使用 XML、RDF 和 OWL 等语义 Web 技术来表示云上的数据。需要注意到，大部分工作都侧重于挖掘关系数据。最近，出现了有关于挖掘非结构化数据（如文本、音频、图像和视频）的工作。挑战是挖掘存储，以及管理 XML 和 RDF 文档的数据库。另一方面是挖掘 XML 和 RDF 文档，而不会透露实际数据，而是提供相关性和趋势。前者是数据挖掘的一个方面，而后者则是基于隐私保护的数据挖掘的一个方面。还有第三个方面，就是用本体来帮助挖掘过程。例如，数据挖掘工具可能需要关于云页面的含义的说明。在这里，OWL 中表达的本体可用于阐明促进挖掘过程的概念（见图 30.3）。

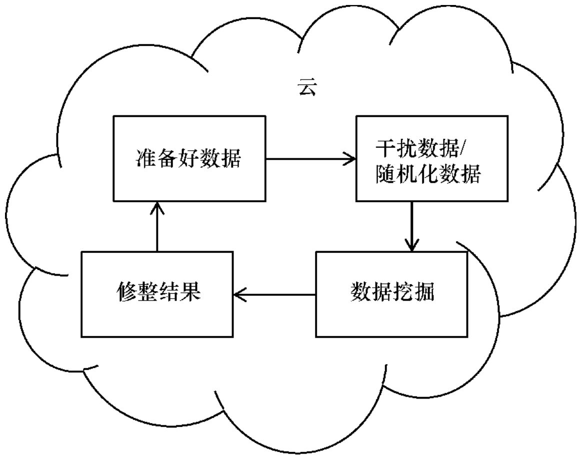


图 30.3　保护隐私的云数据挖掘

30.3　隐私管理和云

30.3.1　云服务的隐私管理

隐私管理可以作为云服务的集合来实现。例如，当用户从云端请求资源时，会调用用于隐私管理的云服务。该服务将向用户呈现 CSP 的隐私策略，用户随后可以确定是否请求资源。图 30.4 说明了用于隐私管理的云服务，而图 30.5 为用于隐私管理的语义云服务。



图 30.4　保护隐私的云服务

语义 Web 技术对隐私管理的主要贡献是在语义 Web 技术中指定策略。这些策略可以用 XML、RDF、OWL 或相关的语义 Web 语言实现。另一个贡献是 P3P。CSP 可以利用这些语义 Web 技术来表达和推理隐私策略。

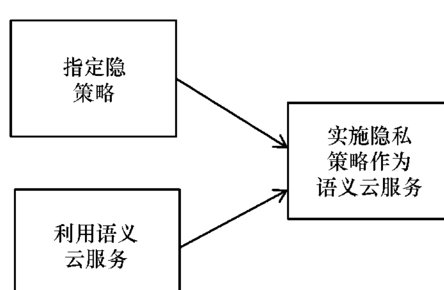


图 30.5　保护隐私的语义云服务

如信任管理一样，需要确定适当的语言来指定隐私策略。为此，XML 正在成为流行的语言。即使最初专注于将 RDF 用于隐私策

略规范的 P3P 标准，也转换为 XML。但是，如果需要隐藏隐私策略的语义和隐私的理由，则 RDF 或 OWL 将更为合适。

在规定隐私策略时，还需要确定是否泄露敏感信息或私人信息。所以，可能对原始隐私策略本身，实施适当的机密性或隐私策略。因此，我们可能希望对隐私策略描述中的各个部分进行访问控制。

30.3.2 云服务和语义云服务中的隐私

云服务中的隐私就是关于保障云服务的隐私性。例如，云服务可能正在处理高度敏感的信息或进行监视。另外，私人和敏感信息，例如社会保障号码和/或个人的位置，可能必须通过适当的策略执行来保护。图 30.6 所示为云服务中的隐私。



图 30.6　云服务中的隐私

语义云服务中的隐私主要在于确保私有信息不被这些语义云服务泄露。需注意，语义云服务是利用基于云的 WS 和语义 Web 技术，来表示和推理云数据和策略的服务。我们的目标不是揭露私人信息。为此，我们需要确保在 XML 和 RDF 文档以及 OWL 本体上正确执行隐私策略。更进一步，基于描述逻辑开发的推理引擎的目的，就是使得私人信息不能通过演绎来推断。

云服务的隐私很少受到关注。Bertino 及其同事对 XML 的隐私进行了研究，并且还研究了基于 Trust - X 系统的信任管理产生的侵犯隐私行为（见本章参考文献［SQUI07］）。Finin 等人正在为他们的语义 Web 研究隐私，尽管他们的研究主要集中在信任管理上。在我们对语义云服务的 CPT（机密性、隐私和信任）的研究中，基于我们所说的基本系统和高级系统进行隐私保护（见本章参考文献［THUR06a］）。需注意的是，高级系统由隐私引擎组成，通过推理将重点关注隐私侵权。利用语义云服务，这个想法是让机器检查云中的语义数据，并确定是否显示任何私人信息。此外，在普通云中，CSP 将向用户显示其隐私策略，用户决定是否将他/她的私人信息存储在云中。使用语义云服务，CSP 可以检查隐私策略和用户偏好，并也可以向用户提供他/她应该存储私人信息的建议。图 30.7 所示为语义云的隐私管理。

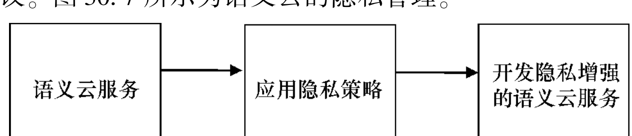


图 30.7　语义云服务中的隐私

30.4 总结和展望

在本章中，我们讨论了各种隐私概念，并提供了隐私管理的概述。然后，讨论了隐私管理和云服务。例如，云服务必须维持隐私。隐私控制器可以作为云服务被实现。我们还讨论了用于指定隐私策略的语义云中的隐私。

本章的大部分讨论都处于研究的早期阶段。我们没有尝试讨论隐私的正确定义。我们的目标是说明隐私管理与语义云之间的联系。正如我们所提到的，语义 Web 技术在隐私策略的规范和推理中是有用的。此外，我们已经讨论了由 XML 和 RDF 表示的数据可以被挖掘，因此这可能侵犯个人的隐私。

我们在工作中强调，只有技术是不足以保护个人隐私的。我们需要社会科学家、技术专家和决策者共同合作。引进法律专家也很重要。有人说，不可能防止侵犯隐私，法律措施是唯一可行的解决方案。不过，我们认为“有一些隐私比没有好”，但是我们必须小心，不要造成虚假的隐私或安全隐患。

参 考 文 献

[P3P] Platform for privacy preferences, www.w3c.org

[SQUI07] Squicciarini, A. C., E. Bertino, E. Ferrari, F. Paci, and B. Thuraisingham, PP-Trust-XA system for privacy preserving trust negotiations. ACM TISSEC—*Transactions on Systems and Information Security*, 10(3), Article 12, 2007.

[THUR90] Thuraisingham, B., Recursion theoretic complexity of the inference problem, *Computer Security Foundations Workshop*, 1990 (also, Technical report, The MITRE Corporation, MTP-291), Franconia, NH.

[THUR05] Thuraisingham, B., *Database and Applications Security: Integrating Data Management and Information Security*, CRC Press, Boca Raton, FL, 2005.

[THUR06a] Thuraisingham, B., N. Tsybulnik, and A. Alam, Administering the semantic web: Confidentiality, privacy and trust management. *Journal of Information Security and Privacy*, 1(1): 18–34, 2007.

[THUR06b] Thuraisingham, B., On the complexity of the privacy problem, Technical report, The University of Texas at Dallas, 2006.

[W3C] World Wide Web Consortium: www.w3c.org

第 31 章　完整性管理、数据来源和云服务

31.1　概述

在本章中，我们将讨论云服务的完整性管理。完整性包括几个方面。在数据库世界中，完整性包括并发控制和恢复，以及执行完整性约束。例如，当同时执行多个事务时，必须确保数据的一致性。当事务中止时，必须确保数据库从故障恢复到一致的状态。完整性约束是数据必须满足的规则。这些规定包括“薪资必须是正的”及“员工的年龄不能随时间而减少”等。最近，完整性又包括了数据质量、数据来源、数据普及性、实时处理和容错性。

在本章中，我们将讨论云服务的完整性方面，以及将完整性管理作为云服务来实现。例如，我们如何确保数据和流程的完整性？我们如何确保数据质量得到保持？本章的结构如下。在 31.2 节中，我们将讨论完整性、数据质量和数据来源方面的内容。其中完整性方面将在 31.2.1 节进行讨论。数据质量和数据来源将在 31.2.2 节讨论。31.3 节讨论云服务和完整性管理。数据完整性和云服务的来源将在 31.3.1 节中进行专门讨论。31.3.2 节将讨论云服务的数据完整性。在 31.4 节对本章进行总结。完整性的各个方面如图 31.1 所示。

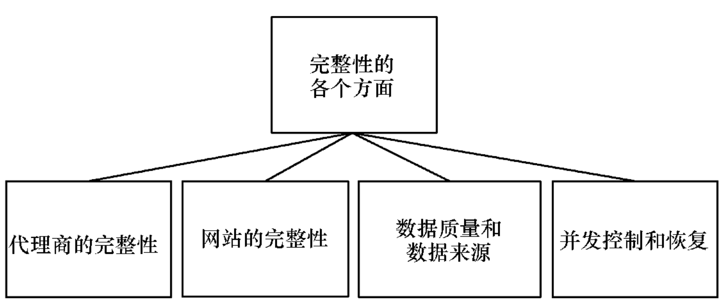


图 31.1　完整性的各个方面

31.2　完整性、数据质量和数据来源

31.2.1　完整性方面

完整性有很多方面。例如，并发控制、恢复、数据准确性、满足实时约束、数据准确性、数据质量、数据来源、容错和完整性约束执行，这些都是完整性管理的一个方

面。如图 31.1 所示。在本节中，我们将研究完整性的各个方面。

并发控制：在数据管理中，并发控制是关于同时执行的事务，并确保数据的一致性。因此，当多个事务尝试同时访问数据时，事务必须获取锁或利用时间戳来确保数据保持一致状态。在集中式和分布式环境下，对事务管理的并发控制技术进行了广泛的研究（见本章参考文献［BERN87］）。

数据恢复：当事务在完成执行之前中止，数据库应该恢复到一致的状态，例如事务开始执行之前的状态。目前，已经提出了几种恢复技术来确保数据的一致性。

数据真实性：当数据传送给用户时，必须保证其真实性。也就是说，用户应该获得准确的数据，数据不应该是被篡改的。我们在第三方发布时进行了确保 XML 数据真实性的研究（见本章参考文献［BERT04］）。

数据完整性：用户收到的数据不仅应该是真实的，而且还应该是完整的。也就是说，用户被授权看到的所有内容都必须传递给用户。

数据现势性：数据必须是当前的。也就是说，过期的数据必须被删除或存档，并且用户看到的数据必须是当前数据。数据现势性是实时处理的一个方面。如果用户想要检索温度，则必须给出当前温度，而不是 24 小时前的温度。

数据准确性：也就是数据的准确度如何？这也与数据质量和数据现势性密切相关。也就是说，准确性取决于数据是否被恶意破坏，或者是否来自不受信任的来源。

数据质量：数据质量高吗？这包括数据真实性、数据准确性，以及数据是否是完整的或确定的。如果数据不确定，那么我们可以在不确定性的前提下，确保使用数据的操作不受影响吗？数据质量也取决于数据源。

数据来源：这与数据的历史有关，也就是说，从数据产生（比如从传感器发出数据）的时间开始，直到这些数据被交给决策者的当前时间。问题是谁访问过数据？谁修改了数据？数据又如何运行？这将确定数据是否被滥用。

完整性约束：这些是数据必须满足的规则，如一个人的年龄不能是负数。这种完整性已被数据库和人工智能社区广泛研究。

容错：如在数据恢复的情况下，失败的进程必须被恢复。因此，容错就是处理数据恢复以及进程恢复。容错技术包括检查点和验收测试。

实时处理：数据现势性是实时处理中一个方面，也就是实时处理中的数据必须是最新的。实时处理也必须满足处理事务所需的时间限制。例如，在 5min 内必须提供股票报价。如果没有，那将来不及了。缺少时间限制可能导致完整性违规。

31.2.2　推论、数据质量和数据来源

一些研究人员认为数据质量是数据来源的应用。此外，他们制定了推断数据质量的理论。在本节中，我们将考察数据质量、数据来源和语义云服务之间的关系。

数据质量是关于数据的准确性、及时性和可靠性（即可信任性）。但是，它是主观的，取决于用户和域。必须回答的一些问题中包括创建数据，即它们从哪里来？为什么要获取数据以及如何获得数据？数据质量信息作为注释存储在数据中，应该是数据来源

的一部分。人们可以问我们如何获得数据的可信度。这可能取决于数据源的排名和声誉。需要注意的是，我们在第29章中讨论了声誉。

正如我们所说，研究人员已经开发了推断数据质量的理论（见本章参考文献[PON]）。动机是因为数据可能来自多个数据源，它们是共享的、容易出错的。此外，数据可能不确定。因此，统计推理、贝叶斯理论和Dempster－Schafer证据理论等不确定性理论，被用来推断数据的质量。关于安全性，我们需要确保所推断数据的质量不违反策略。例如，在未分类的级别，我们可以说数据源是可信赖的，但在机密级别，我们知道数据源是不可靠的。我们开发的推理控制器，可以与为数据质量开发的推理理论相结合，以确保安全性。

接下来，我们来看看数据来源。对于包括医疗和保健在内的许多领域，以及数据准确性至关重要的防御，我们需要对数据来自何处以及谁可能篡改数据有很好的了解。如本章参考文献［SIMM05］所述，数据来源是一种有时称为“家系”或“谱系”的元数据，也是描述一个数据的来源以及其到达数据库的过程。数据来源是一种信息，能有助于确定数据产品从原始来源开始派生的历史。

来源信息可以应用于数据质量、审计和所有权等。通过记录谁访问数据，可以确定数据是否被滥用。通常，注释用于描述与数据相关的信息（例如，谁访问数据？数据来自哪里？）。挑战在于确定是需要维护粗粒度来源数据，还是细粒度来源数据。例如，在粗粒度情况下，可以注释关系的表。而在细粒度的情况下，就可以注释每个元素。当然，要考虑管理来源的存储开销。XML、RDF和OWL已被用于表示来源数据。这样，为语义Web技术开发的工具可用于管理来源数据。

使用数据来源进行滥用检测有很大的好处。例如，通过维护数据的完整历史记录、访问数据的人员、访问数据的时间和地点，可以回答如“2010年1月至5月期间谁访问数据”的查询。因此，如果数据是损坏的，可以确定谁损坏了数据或数据何时被损坏。图31.2所示数据来源的内容。我们对使用语义Web技术表示策略和来源数据的表示和推理进行了广泛的研究（见本章参考文献［CADE11a］、［CADE11b］）。

数据来源

谁创建了数据?

数据来自哪里?

谁访问了数据?

数据的完整历史是什么?

数据被滥用了吗?

图31.2　数据来源

31.3　完整性管理和云服务

31.3.1　用于完整性管理的云服务

这里有两个方面。一个是可以使用云服务实现完整性管理。另一个是确保云服务具有高度完整性。为了实现作为云服务的完整性管理，方法是调用云服务以确保数据质量，以及数据和系统的完整性。图31.3所示为将完整性管理作为云服务实现的情况。

像保密性、隐私和信任一样，如XML和RDF之类的语义Web技术也可以用于指定

完整性策略。完整性策略可以包括用于指定完整性约束的策略，以及用于指定时序约束、数据现势性和数据质量的策略。以下是策略的一些示例：

完整性约束：员工的年龄必须是正的。在关系表示中，可以将此策略指定为

EMP.AGE > 0.

In XML, this could be represented as the following:

```
<Condition Object=“//Employee/Age”>
  <Apply FunctionId=“greater-than”>
    <AttributeValue  DataType=“http://www.w3.org/2001/XMLSchema#integer”>0
    </AttributeValue>
  </Apply>
</Condition>
```



图 31. 3　用于完整性管理的云服务

数据质量策略：员工表中数据的质量为低。在关系模型中，这可以表示为 EMP. Quality ＝ LOW。

在 XML 中，此策略可以表示为

```
<Condition Object=“//Employee/Quality”>
 <Apply FunctionId=“equal”>
   <AttributeValue DataType=“http://www.w3.org/2001/XMLSchema#string”>
LOW
   </AttributeValue>
 </Apply>
</Condition>
```

数据现势性：一个例子是 EMP 的工资值不能超过 365 天。在关系表示中，这可以表示为

AGE（EMP. SAL）＜＝ 365 天

在 XML 中，这表示为

```
<Condition Object=“//Employee/Salary”>
  <Apply FunctionId=“AGE”>
    <Apply FunctionId=“less-than-or-equal”>
      <AttributeValue DataType=“http://www.w3.org/2001/XMLSchema#integer”
>365
      </AttributeValue>
    </Apply>
  </Apply>
</Condition>
```

以上示例显示了如何指定某些完整性策略。需要注意的是，语义 Web 技术还有许

多其他应用可以确保完整性。例如，为了确保数据来源，必须存储数据的历史记录。我们正在使用语义 Web 技术（如 XML）来表示用于确定数据质量或数据是否被滥用的数据注释。也就是说，捕获的数据用元数据信息进行注释，例如，数据是什么？何时被捕获？是谁捕获的？然后，随着数据从一个地方移动到另一个地方，或从一个人到另一个人，这些注释都将被更新，使得在稍后的时间，可以分析数据是否被滥用。这些注释通常在语义 Web 技术（如 XML、RDF 和 OWL）中表示。

用于完整性管理的语义 Web 技术的另一个应用，是使用本体来解决语义异构。也就是说，语义异构可能导致破坏完整性。当同一个实体在不同的站点被认为是不同的，从而破坏完整性和准确性时，就会发生这种情况。比如，通过使用 OWL 中指定的本体，可以表示在一个站点中的船舶和另一个站点中的潜艇是一样的。

语义 Web 技术在推论和推理方面也有应用。例如，可以使用基于 RDF、OWL 或比如规则的推理引擎，来确定是否违反了完整性策略。在这部分中，我们讨论了推理和隐私问题，以及建立推理引擎。还必须对违反完整性的策略进行研究。在语义云服务的情况下，调用这些服务来强制或确定数据是否已损坏。图 31.4 所示为语义云服务在完整性管理中的应用。

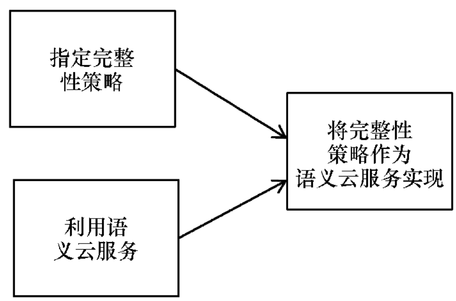


图 31.4　完整性管理中的语义云服务

31.3.2　云和语义云服务的完整性

虽然我们讨论了使用云服务和语义云服务来进行完整性管理，但在本节中，我们研究确保云服务具有高度完整性的问题。这里的想法是确保云服务不是恶意的，不会破坏数据或其他服务。这意味着云服务必须经过验证或测试，不包含恶意代码。图 31.5 所示为云服务自身的完整性管理。



图 31.5　云服务自身的完整性

我们还需要确保语义 Web 技术的完整性得到维护。用于数据质量和来源的注释通常用 XML 或 RDF 文档表示。这些文件必须准确、完整和保持最新。因此，语义云服务必须确保文档保持完整性。完整性的另一个方面是管理由 XML 或 RDF 文档组成的数据库。这些数据库有关于关系数据库的所有问题和挑战。也就是说，必须优化查询，且并发执行事务。因此，XML 和 RDF 文档的并发控制和故障恢复，成为管理 XML 和 RDF 数据库的一个挑战。在这种情况下，语义云服务要执行安全的数据管理功能。

作为语义云服务被实施的代理，可执行搜索、查询和集成异构数据库等操作，也必

须确保数据的完整性得到维护。这些代理不能恶意地破坏数据，他们必须确保数据准确、完整和一致。最后，当集成异构数据库时，人们正在使用如 OWL 本体的语义 Web 技术来处理语义异质性。这些本体也必须是准确和完整的，且不能被篡改。

总之，为了使语义 Web 技术变得有用，它们必须保护完整性。此外，语义 Web 技术本身正被用于指定和推理完整性策略。图 31.6 所示为语义云服务的完整性管理。



图 31.6　语义云服务的完整性

31.4　总结和展望

在本章中，我们对数据完整性进行了概述，包括数据质量和数据来源，讨论了语义 Web 技术在数据完整性方面的应用，并讨论了语义 Web 技术的完整性。最后，我们概述了数据质量与数据来源之间的关系。

数据来源和数据质量虽然重要，但最近才受到重视。这是由于云上有大量的信息，知道数据的准确性以及数据是否被复制或剽窃都很重要。我们还需要回答有关拥有数据的问题，数据被滥用？因此，数据来源对于确定数据的安全性至关重要。

云服务应具有较高的完整性。此外，完整性技术可以作为云服务实现。语义 Web 技术提供了一种表示和存储数据质量和数据来源的方法。随着我们在这些技术方面取得的进展，我们将为数据质量和数据来源管理提供改进的解决方案。本质上，数据质量和数据来源是数据安全的一部分，语义 Web 技术对于管理数据质量和数据来源信息非常有用。

参 考 文 献

[BERT04] Bertino, E., B. Carminati, E. Ferrari, B. Thuraisingham, and A. Gupta, Selective and authentic third party distribution of XML documents, *IEEE Transactions on Knowledge and Data Engineering*, 2004, 16(10), 1263–1278.

[BERN87] Bernstein, P. et al., *Concurrency Control and Recovery in Database Systems*, Addison-Wesley, MA, 1987.

[CADE11a] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B.M. Thuraisingham, A language for provenance access control. CODASPY2011: 133–144.

[CADE11b] Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. M. Thuraisingham, Transforming provenance using redaction. SACMAT2011: 93–102.

[PON] Pon, R. K. and A. F. Cárdenas, Data quality inference, UCLA Report.

[SIMM05] Simmhan, Y. L., B. Plale, and D. Gannon, *A Survey of Data Provenance in e-Science*; Indiana University Technical Report, also in SIGMOD record, 34(3), 2005.

第八部分总结

虽然以前的部分主要关注访问控制模型和云服务的机密性，但在第八部分中，我们讨论了云服务的信任、隐私和完整性问题。特别是，我们讨论了隐私控制、数据质量以及管理信任。机密性、信任性、隐私性和完整性等特征如果包含到云中，将带来可靠的云。

在第 29 章中，我们讨论了信任管理，及其与云服务和语义云服务的连接。我们首先讨论了信任管理的各个方面，包括定义信任和描述信任协商。然后，讨论了在语义云服务的环境中实施信任的内容。我们还讨论了使用语义云服务技术来指定信任策略。最后，论述了包括风险信任管理和声誉网络在内的相关概念。

在第 30 章中，我们讨论了隐私的各种概念，对隐私管理进行了概述。然后，讨论了云服务的隐私管理。例如，云服务必须保持隐私性。此外，隐私控制器可以被实现为云服务。我们还讨论了语义云的隐私和语义 Web 技术的使用，以指定和推理隐私策略。

在第 31 章中，我们提供了数据完整性的概述，包括数据质量和数据来源。接着讨论了语义云服务在数据完整性方面的应用，并论述了语义云的完整性。此外，我们还概述了数据质量与数据来源之间的关系。

第九部分　面向云安全建立基础设施、教育项目和研究计划

第九部分简介

现在我们已经概述了在云计算和安全的云计算中开发的系统，我们将描述正在开发的基础设施、教育项目和研究计划，这是我们在云计算安全工作的基础。

第九部分由 3 章组成：第 32 章、第 33 章和第 34 章。第 32 章将讨论我们在 UTD 开发的云计算安全基础架构。第 33 章讨论我们在云计算安全中的教育项目。在第 34 章中讨论过去五年里我们主导的合作研究计划。我们在前几章中讨论的很多原型都是由我们的研究计划产生的。这些原型不仅帮助我们建立安全的基础设施和提升我们的研究，而且还帮助我们在安全云计算中建立了强大的教育项目。

第 32 章　云安全的基础设施

32.1　概述

正如我们之前讨论的一样，如今迫切需要安全存储、管理、共享和分析大量复杂的（例如半结构化和非结构化）数据来确定模式和趋势，以提高医疗保健质量，更好地维护国家安全，并探索可替代能源。但是，据我们所知，没有现成的基础设施可以满足上述需求。因此，我们正在 UTD 开发一个查询大量复杂（例如语义 Web 和地理空间）数据，并维护这些数据的机密性和隐私性的基础设施。

新兴的云计算模式试图解决网络连接设备的增长，并处理大量数据的问题。Google 引进了 Map/Reduce 框架来处理硬件商品上的大量数据。而 Apache 的 HDFS 正在成为云计算的高级软件组件，当然也结合了 Map/Reduce 等集成部件。增加人类推理、解释和决策能力促使了语义 Web 的出现，这是一种将网络从目前只有人类可读的形式，转变为机器可处理形式的尝试。这反过来又导致了大量的社交网络站点的出现，它们拥有大量的数据可供共享和管理。因此，我们迫切需要一个可以扩展到处理大量站点和大量数据的基础设施。然而，使用 HDFS 和 Map/Reduce 中最先进技术的基础设施是不够的，因为①它们没有提供足够的安全机制来保护敏感数据，②它们也没有能力处理大量语义 Web 和地理空间的数据。我们正在利用最先进的硬件、软件和数据组件，构建一个处理当前云计算基础架构不足之处的基础架构。特别的，我们使用流行硬件部件（例如安全协处理器）来提高性能，因为它包含额外的安全功能；集成开源软件部件和定制开发的软件部件，以支持对复杂数据的查询操作；支持细粒度的访问控制和引用监视器支持，使其为复杂数据提供安全性；为云计算提供强大的认证机制。该基础设施有助于计算机和信息科学，以及社会科学等几个领域的科学研究，包括安全和隐私、语义 Web、地理空间信息和社交网络分析。

我们工作的技术贡献是支持细粒度的云访问控制、加密的敏感数据的存储、用于海量数据集的复杂查询处理和认证机制。我们首先开发了此类云，能够面向应用，比如使用大量数据时的信息共享保障，它由最先进的硬件、软件和数据组件组成。我们的研究项目用于语义 Web 数据管理、信息共享保障和自动引用监视器等。应该指出，我们的基础设施的开发正在 AFOSR 和 NSF 的资助下进行。

本章结构如下：在 32.2 节中，我们将论述基础设施。在 32.3 节讨论与其他基础设施相集成。32.4 节讨论我们基础设施的研究改进。32.5 节讨论教育项目和教育实施。在 32.6 节对本章总结。图 32.1 所示为本章的内容。图 32.2 所示为基础设施的组件。

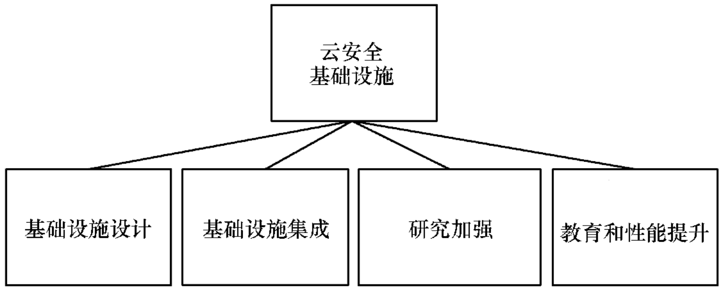


图 32.1 云安全基础设施

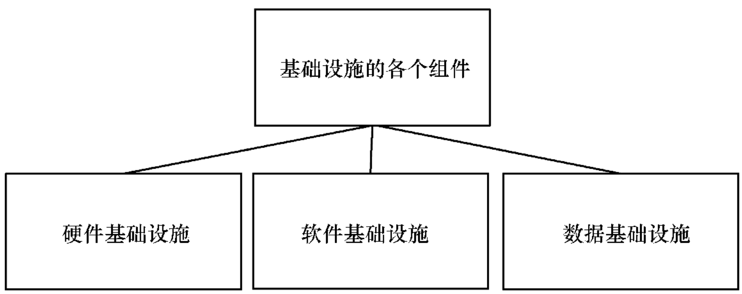


图 32.2 基础设施的各个组件（来自 Hamlen，K. W.，Kantarcioglu，M.，和 Khan，L 的论文“Security Issues for Cloud Computing”，发表自 International Journal of Information Security and Privacy，Namati，H.（ed）.，4（2），36－48. © 2010，IGI Global. 已获得授权）

32.2 基础设施研究描述

32.2.1 背景

32.2.1.1 我们基础设施的需求

现在出现了安全地存储、管理、共享和分析大量数据的需求。例如，我们会想统计分析多年的股票市场数据，来揭示一个模型，或者根据几年的天气和相关数据建立可靠的天气模型。为了处理在许多站点（即节点）上分布的大量数据，我们需要具有可扩展硬件和软件组件的基础设施。云计算模型的出现，用以解决网络连接设备的爆炸性增长和处理大量数据的问题。大规模的可扩展性和新的互联网驱动的经济学是它的定义和特征。Hadoop 正在成为云计算高级软件的解决方案，它需要集成 Mahout、Hama 和 Map/Reduce 等的软件部分（见本章参考文献［MAHO］、［HAMA］、［CHU07］、［DEAN04］）。惠普 Open Cirrus 测试台等基础设施用的是 HDFS。虽然这些基础设施具有 Hadoop 带来的优势，但是它们也具有 Hadoop 的局限性，包括无法处理复杂数据，也没有足够的安全保护。由于没有可以处理由语义 Web 和地理空间数据组成的 PB 级数据集的基础设施，以及为这些数据提供存储和访问的安全保障，因此我们正在开发相关的基础设施。我们的基础设施包括硬件组件、软件组件和数据组件。

32.2.1.2　云计算中的 Hadoop

我们基础设施软件组件的主要部分是 HDFS，它是一种分布式的基于 Java 的文件系统，有能力存储 PB 级数据的大量节点。理想情况下，文件大小是 64MB 的倍数。它通过在多个主机上复制数据来实现可靠性。默认复制值是 3（即数据存储在 3 个节点上）。其中两个节点驻留在同一个机架上，而另一个节点位于不同的机架上。一组数据节点集群构建文件系统。节点通过 HTTP 传输数据，客户端使用 Web 浏览器访问数据。数据节点相互通信以调节、传输和复制数据。

HDFS 架构基于主从式结构（见图 32.3）。主站被称为 Namenode 并包含元数据。它保留了所有文件的目录树和集群中节点的可用数据的记录。该信息作为图像存储在存储器中。数据块存储在 DataNode 中。Namenode 是故障单点，因为它包含元数据。因此，可以在任何机器上设置可选的辅助 Namenode。客户端访问 Namenode 以获取所需文件的元数据。获取元数据后，客户端直接与各自的 Datanode 进行对话，以获取数据或执行 IO 操作（输入/输出）（见本章参考文献［HADO］）。Map/Reduce 引擎存在于文件系统之上。该引擎由 JobTracker 组成。客户端应用程序向该引擎提交 Map/Reduce 作业。Job Tracker通过将作业推送到集群中的可用任务跟踪器节点，来将作业放置在数据附近。

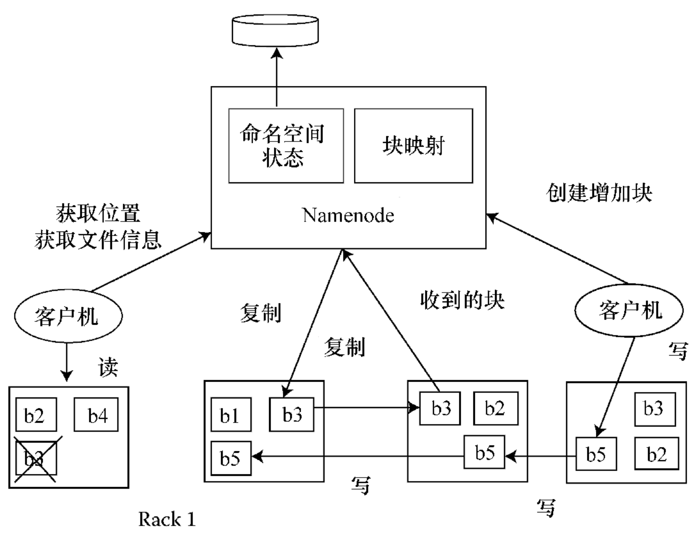


图 32.3　HDFS 架构

32.2.1.3　Hadoop 的不足

在撰写本书的时候，正在使用的 Hadoop 的基础设施有以下局限性：

1）无法处理加密的敏感数据：从医疗记录到信用卡交易等敏感数据，需要使用加密技术存储以进行额外的保护。目前，HDFS 不支持对加密数据来执行安全而有效的查询处理。

2）语义 Web 数据管理：需要一种可行的解决方案，来提高对如 RDF 之类的语义 Web 数据的查询性能和可扩展性。RDF 数据集的数量正在增加。存储数十亿 RDF 三元组的问题和如何对其进行有效查询的问题尚待解决（见本章参考文献［MUYS06］、［TESW07］、［RAMA09a－c］）。目前，不支持在 HDFS 中存储和检索 RDF 数据。

3）非细粒度的访问控制：HDFS 不支持细粒度访问控制。有一些工作可以为 HDFS 提供访问控制列表（见本章参考文献［ZHAN09］）。对于像信息共享保障等许多应用，访问控制列表是不足够的而且需要支持更复杂的策略。

4）没有强大的认证：连接到 Jobtracker 的用户，只要拥有设置 HDFS 的账户权限，就可以提交任何作业。HDFS 的未来版本将支持网络认证协议。例如，Kerberos 用于用户认证和数据传输的加密（见本章参考文献［ZHAN09］）。但是，对于一些有保障的信息共享方案，我们需要 PKI 提供数字签名支持。

32.2.2　基础设施开发

我们正在开发云计算安全的基础设施。它由机械磁盘驱动器上的硬件组件（包括 800TB）的数据存储器，2400GB 的内存和 100 个商用计算机、一个软件组件（包括 Hadoop）和一个数据组（包括语义 Web 数据存储库）。这个基础设施为研究人员提供以下支持：①加密敏感数据的高效存储；②存储、管理和查询大量的数据；③细粒度访问控制；④强认证。基础设施的开发由四部分组成，以解决 HDFS 的每个局限性（如图 32.4 所示）：

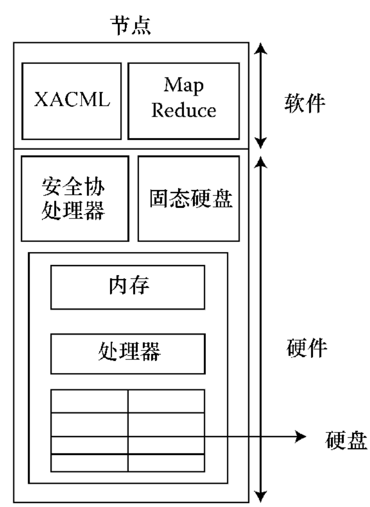


图 32.4　安全协处理器

1）将安全协处理器（Secure Coprocessor，SCP）部件并入硬件组件中来实现敏感数据的高效加密存储。

2）将包括 Mahout（见本章参考文献［MAHO］）、Hama（见本章参考文献［HAMA］）、Jena（见本章参考文献［JENA］）和 Pellet（见本章参考文献［PELL］）在内的软件组件的几个部分整合在一起，并开发出用于通过 HDFS 进行 SPARQL 查询处理的部件。

3）将 XACML（见本章参考文献［MOSE05］）实现部分整合到软件组件中，实现细粒度访问控制。

4）为灵活认证机制引入了一个组件。

应注意到，条目 2）~4）是软件组件的一部分，而是 1）是硬件组件部分。

可能会有人问我们这个问题，为什么不在目前的云计算基础架构，如 Open Cirrus 提供的硬件组件上实现软件组件？我们已经尝试了这个选项。首先，Open Cirrus 根据其经济模式（例如，虚拟现金）会提供有限的访问。此外，Open Cirrus 不提供我们需要的硬件支持（例如，安全协处理器）。不过，我们正在与 HP 讨论将我们的基础设施整

合到 Open Cirrus 上。

32.2.3 基础设施的硬件组件

32.2.3.1 硬件组件的集群部分

在 UTD，我们已经拥有大量的硬件来支持我们的研究。这个硬件是基础设施硬件组件的一部分。我们目前的硬件包括 3 个具有不同配置的主要集群。第一个集群的体积非常小，通常用作我们的测试集群。它由 4 个节点组成，每个节点都有 Pentium－IV 机器，硬盘空间为 80GB，主内存为 1GB。我们使用此集群中的样本数据来测试我们的代码并执行各种优化算法。第二个集群放置在 SAIAL（有实验室支持的安全分析和信息保障实验室，Security Analysis and Information Assurance Lab with lab support），共有 32 个节点。此集群中的所有节点都运行在 Hadoop 的商品硬件上。这个 32 节点集群有一个混合的硬件集合，10 个节点具有一些高端配置，如 Pentium－IV 机器，每个磁盘空间为 500GB，主机内存为 4 GB。剩下的 22 个节点也是每个具有磁盘空间大约为 80GB，2GB 主存的 Pentium－IV 机器。所有这些节点通过内部网络上的 48 端口 Cisco 交换机彼此连接。只有主节点才能从公网访问。我们可以访问的第三个集群是 HP 实验室的 Open Cirrus 基础设施测试台。我们可以从他们的测试台使用多达 30 个节点。每个节点均具有高端配置，如具有 8GB 主存储器和超过 1TB 硬盘空间以及四核处理器。

我们考虑了项目的 3 种可能的配置。如下：

1）戴尔精密机器（商品硬件，但每台机器存储空间较小）。

2）来自不同提供商的，且可以支持更多磁盘的组装服务器机器（商品硬件）。

3）用于数据存储的 SAN 存储和用于计算的 90 台计算机（成本为 300 万～400 万美元，其针对 PB 级数据）。

由于方案 3 特别昂贵，我们没有进一步研究这个方案。因此，我们进一步探索了方案 1 和方案 2。

方案 1：Dell 机器配置－集群中的每个节点实例都是戴尔精密机器，拥有 4.8GT/s 的单个四核英特尔至强处理器 E5502，它有 24 GB DDR3 主内存，1066MHz 和 6 DIMM。这样的配置允许我们使用每核心高达 6GB 的内存。这些节点中的每个节点均可以存储到最多 4 个硬盘，每个硬盘 1.5TB，每个节点扩大到 6TB 存储。考虑到 Hadoop 的复制因子为 3，每个节点实际上可以存储最多 2TB 的数据。因此，如果我们选择了这些机器来扩展云和存储 800TB 的数据，那么我们就需要 400 台机器，这导致物理存储将是一个问题。

方案 2：组装的服务器配置－集群中每个节点的一个实例是 2U 机架式服务器。每个包含两个四核英特尔至强处理器 5500 系列，快速通道互联（Quick Path Interconnect，QPI）高达 6.4GT/s。此外，每个服务器包含 24 GB（12×MEM DDRIII 1333 2 GB ECC/REG）内存。因此，这种配置允许我们使用每个处理器大约有 12GB 的内存和每个内核有 3GB 的内存。我们也有空间将内存扩展到 48GB。此外，每个服务器都配有 Infband 20Gbit/s 控制器，其中的 SATA 还支持硬盘 RAID 5。网络控制器是 Intel® 82576 双端口千兆以太网控制器，并且也支持 10BASE－T、100BASE－TX 和 1000B ASE—T、RJ45 输

出，通过网络进行 IO 操作或数据传输时，可以减少出现瓶颈的可能性。Hadoop 是分布式文件系统，因此，每个节点都是固态驱动器以及传统存储设备组合的设备。每个服务器均可能包含 128GB 固态磁盘（SSD）以及 12 个具有传统存储功能的磁盘。它具有 12 ×3.5“热插拔 SAS/SATA 驱动器托盘，我们正在使用的 12 × HD WD20EADS SATA2 2TB 低功耗版本，为每个节点提供总共 24TB 的物理可用磁盘空间。Hadoop 是可靠的，因为它维护数据的复制，默认复制因子为 3，每个节点可以存储最多 8TB 的数据，其中 5% ~10% 由 Hadoop 临时目录使用，需要交换空间用于 Map – Reduce 功能。

这些工作一旦完成，我们的云将由最多 5 个机架中的 100 个这样的节点组成。这些节点是存储节点和计算节点的组合，其中一些两者兼有。我们正在为机架内通信提供 Cisco 6509 交换机，以提高吞吐量。另外，根据上述每个节点的配置，我们拥有 2400TB（2.4PB）的传统磁盘存储空间和约 3.2TB 的 SSD（假设部署了 100 个节点的 25 个 SSD）。考虑到 Hadoop 的复制因子，我们的云最多可以存储高达 800TB 的数据。

接下来，我们将 HDFS 规范与基准硬件进行了比较。Hadoop – Apache Wiki 上的基准硬件提供了两个名为 Herd 1 和 Herd 2 的基准集群。例如，Herd 1 中的每个节点包括 Intel Xeon LV 2.0GHz，四核，4GB RAM 和 4 个磁盘，每个均是 0.25TB SATA。Herd 1 由两个机架中的 35 个节点组成。在我们的云中，每个节点都有两个四核处理器和大容量的 RAM（24GB），比基准硬件要好。因此，我们预计应用在我们的云上会有很好的效果。

32.2.3.2　硬件组件中的安全协处理器部分

我们正在如下的硬件上实现这个组件。嵌入 SCP 来有效处理加密数据（见图 32.5）。基本上，SCP 是能够进行有限的通用计算的防篡改硬件。例如，IBM 4758 密码协处理器（见本章参考文献［IBM04］）是由包含在防篡改外壳中的 CPU、内存和专用密码硬件组成的单板计算机，经 FIPS PUB 140 – 1 认证为 4 级。当安装在服务器上时，它能够进行本地计算且能完全隐藏在服务器中。如果检测到被篡改，则 SCP 会清除内部记忆。由于 SCP 是防篡改的，可能会诱使在 SCP 上运行整个敏感数据存储服务器。将整个数据存储功能推送到 SCP 中是不可行的，原因有很多。

首先，由于防篡改的加壳处理，SCP 通常具有有限的内存（只有几兆字节的 RAM 和几千字节的非易失性存储器）和计算能力（见本章参考文献［SMIT99］）。性能会随着时间的推移而改善，但是像散热/功率使用（必须控制以避免暴露处理）的问题，将迫使通用性和安全计算之间存在差距。另一个问题是在 SCP 上运行的软件必须完全信任和被验证。这个安全要求意味着在 SCP 上运行的软件应该是尽可能地保持简单。那么这个硬件又如何有助于存储大型敏感数据集？我们可以使用随机私钥加密敏感数据集，并减少密钥泄露的概率，我们还可以使用防篡改硬件来存储一些加密/解密密钥（即，加密所有其他密钥的主密钥）。由于密钥不会随时存储在未加密的内存中，攻击者可以通过拍摄系统的快照来寻找密钥。此外，攻击者通过软件或物理机构控制（或篡改）协处理器的任何尝试，都将清除协处理器的内容，这样，也就消除了一种解密任何敏感信息的方法。该框架将促进安全的数据存储和信息共享保障。例如，SCP 可用于保密信息集成，这对保证信息共享非常重要（见本章参考文献［AGRA06］）。

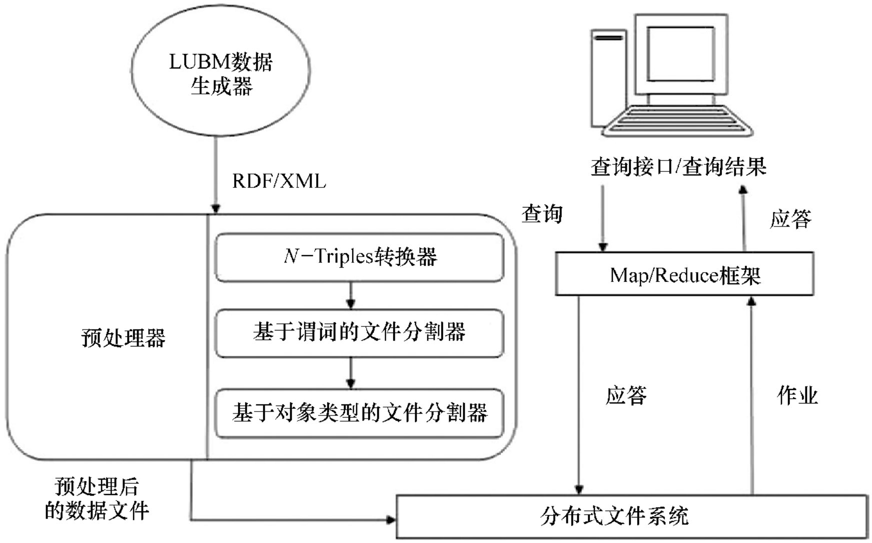


图 32.5　SPARQL 查询处理（来自 Hamlen，K. W.，Kantarcioglu，M.，和 Khan，L 的论文“Security Issues for Cloud Computing”，发表自 International Journal of Information Security and Privacy，Namati，H. （ed.），4（2），36 – 48. © 2010，IGI Global. 已获得授权）

32.2.4　基础设施的软件组件

32.2.4.1　用于存储、查询和挖掘语义 Web 数据的组件部分

软件组件的这一部分包括以下内容：

Jena：这是一个广泛用于解决基于 RDF 数据的 SPARQL 查询的框架（见本章参考文献［JENA］)。但 Jena 的主要问题是可扩展性。它与主存储器的大小成比例。它不支持分布式处理。然而，在我们预处理步骤的初始阶段，我们使用的是 Jena。

Pellet：我们在多个不同的阶段使用 Pellet 推理。我们使用 Pellet 库（见本章参考文献［PELL］）和 Hadoop 的 Map / Reduce 进行实时查询推理功能。

Pig Latin：Pig Latin 是一种运行在 Hadoop（见本章参考文献［GATE09］）之上的脚本语言。Pig 是分析大数据集的平台。Pig 的语言 Pig Latin 可以促进顺序数据的转换。例如，合并数据集并过滤它们，以及将功能应用于记录或记录组。它具有许多内置功能，但我们也可以创建属于自己的用户定义功能来进行专门的处理。使用这种脚本语言，我们可以避免编写 Map / Reduce 代码。可以依靠 Pig Latin 的脚本功能，自动生成脚本代码到 Map / Reduce 代码。

Mahout，Hama：这些是已经增加了 Hadoop（见本章参考文献［MAHO］、［HAMA、［MORE08］）的开源数据挖掘和机器学习软件包。

通过 Hadoop 进行 SPARQL 查询及其优化：SPARQL 是一种查询语言，用于查询 RDF 数据。我们正在开发的软件部分是查询框架，主要查询通过 Hadoop 分发的 RDF 数

据（见本章参考文献［NEWM08］、［MCNA07］、［ROHL07］）。其中有几个步骤来进行预处理和查询 RDF 数据（见图 32.5）。有了这部分，研究人员可以获得结果来优化大量数据的查询处理。下面，我们讨论这一部分开发中涉及的步骤。

预处理：通常 RDF 数据是 XML 格式（参见 LUBM，RDF 数据）。要执行 SPARQL 查询，我们先要执行数据预处理步骤，并将预处理过的数据存储到 HDFS。我们有一个三元组转换器模块，该模块将 RDF/XML 格式的数据转换为更容易理解的 *N* 个三元组格式。为实现这一转换的目的，我们使用 Jena 框架。在基于谓词的文件分割模块中，我们根据谓词拆分所有的 *N* 个三元组格式的文件。因此，数据集的文件总数等于本体/分类法中谓词的数量。在预处理步骤的最后一个模块中，我们在它包含客体类型的基础上进一步划分谓词文件。所以现在，每个谓词文件都有特定类型的客体。这是在 Pellet 库的帮助下完成的。这些预处理数据存储在 Hadoop 中。

查询执行和优化：我们为 Hadoop 开发了一个 SPARQL 查询执行和优化模块（在第 13 章中讨论过）。由于我们的存储策略是基于谓词分割的，所以：第一，我们要查看查询中存在的谓词。第二，并不是查看所有的输入文件，我们只查看输入文件中与谓词匹配的子集。第三，SPARQL 查询通常在其中具有许多连接，并且所有这些连接均可能无法在单个 Hadoop 作业中执行。因此，我们设计一个算法来决定每种查询所需的作业数量。作为优化的一部分，我们应用贪婪策略和基于成本的优化，来减少查询处理时间。贪婪策略的一个例子是覆盖单个作业中可能连接的最大数量。对于成本模型，首先执行的连接是基于汇总统计的（例如，有界变量的选择性因子，为 3 个三元组模式的连接三元组选择因子）。例如，考虑对 LUBM 数据集的查询：

“列出所有属于特定大学校友的人”。

在 SPARQL 中，此查询指定如下：

```
PREFIX rdf: <http://www.w3.org/1999/02/32-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X WHERE {
?X rdf:type ub:Person.
<http://www.University0.edu> ub:hasAlumnus ?X}
```

根据查询中出现的谓词，查询优化器接受此查询输入，决定要查看的输入文件的子集。本体和 Pellet 推理器识别与谓词“hasAlumnus”相关的三个输入文件（在 Under GraduateDegreeFrom，MasterDegreeFrom 和 DoctoraldegreeFrom）。接下来，从文件的类型，我们使用 Pellet 库过滤出客体是人的子类的所有记录。从这三个输入文件（Under Graduate Degree From，Master Degree From 和 Doctoral degree From），优化器根据查询的要求，以及根据 < http ：// www. University 0. edu >过滤出三元组。最后，优化器确定对这种类型的查询中单个作业的需求，然后在该作业中的变量 X 上执行连接。

32. 2. 4. 2 使用 IRM 将 SUN XACML 实现集成到 HDFS 中

目前的 Hadoop 实现强制执行粗粒度的访问控制策略，它实际上将所有系统资源作为一组，进行允许或拒绝访问，而不区分资源。例如，授予访问 Namenode 的用户可以

在任何客户端机器上执行任何程序，并且所有客户机都具有对存储在所有客户端上的所有文件的读写访问权限。当数据、查询和相应的系统资源与安全性相关时，以及并不是所有用户和进程都被完全信任时，这种粗粒度的安全性显然是不可接受的。目前的工作（见本章参考文献［ZHAN09］）是通过实现 Hadoop 的标准访问控制列表，来解决限制对某些系统资源（如文件）访问的问题。然而，这种方法的局限性在于强制执行的安全策略被固化在操作系统中，因此在不修改操作系统的情况下就不容易改变。通过设计 Sun XACML 的 IRM 实现，我们正在对 Hadoop 执行更灵活和更细粒度的访问控制策略。XACML（见本章参考文献［MOSE05］）是 OASIS 标准用于在 XML 中表达访问控制策略的丰富语言。主体、客体、关系和上下文在 XACML 中都是通用和可扩展的，使其非常适合于分布式环境，这种环境下许多不同子策略之间进行交互，以形成更大、复合、系统级的策略。一个抽象的 XACML 执行机制如图 32.6 所示。框架中不受信任的进程，通过向资源的 PEP 提交请求，来访问与安全相关的资源。PEP 将请求重新格式化为策略查询，并将其提交给 PDP。PDP 会咨询任何与请求相关的策略来回答查询。PEP 根据接收到的答案来批准或拒绝资源请求。虽然执行机制的 PEP 和 PDP 组件，传统上是在操作系统级别或可信系统库中实现的，但通过在基础设施中将其实施为 IRM，我们可以实现更大的灵活性。IRM 通过将这些检查直接内插到不可信过程的二进制代码，来实现运行时的安全检查。这具有一个优点，可以在不修改操作系统或系统库的情况下实施该策略。IRM 策略还可以约束在操作系统级别上可能难以或不可能拦截的程序操作。例如，Java 中的内存分配实现时，不调用任何外部程序或库的 Java 字节码指令。

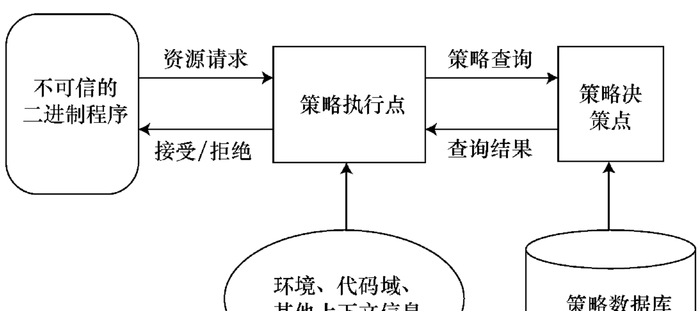


图 32.6 XACML 的执行机制（来自 Hamlen，K. W.，Kantarcioglu，M.，和 Khan，L 的论文“Security Issues for Cloud Computing”发表自 International Journal of Information Security and Privacy，Namati，H.（ed.），4（2），36 – 48. © 2010，IGI Global. 已获得授权）

因此，作为传统的引用监视器，可以在 Java 中执行细粒度的内存绑定策略，因此需要修改 Java 虚拟机或 JIT 编译器。相比之下，IRM 可以识别这些与安全相关的指令，并把适当的保护程序直接注入不可信代码来执行策略。

最后，IRM 可以有效地执行基于历史的安全策略，而不只是限制个人安全相关事件的策略。例如，过去作者在工作中（见本章参考文献［JONE09］）已经使用了 IRM 执行公平性策略，要求不受信任的应用程序共享它们所请求的数据。这可以防止进程受到

基于自由载入行为而产生的拒绝服务攻击影响。由 IRM 注入不可信二进制码的代码，是基于过去的程序操作历史而不是孤立地限制每个程序操作。这涉及将安全状态变量和计数器注入到不可信代码中，但这在操作系统级别难以高效地完成。

IRM 框架的核心包括二进制重写器，它在执行之前静态修改每个不受信任进程的二进制代码，将安全防范机制插入潜在的危险操作中。我们的二进制重写器基于 SPoX（安全策略 XML）实现（见本章参考文献［HAML08］），开发重写器，目的是为 Java 字节码程序执行声明性的基于 XML 的 IRM 策略。为了给我们的系统提供强大的安全保障，我们应用了自动化软件验证技术，包括类型和模型检查，我们以前将它们用于证明二进制重写器的输出（见本章参考文献［HAML06］、［DEVR09］）。这种认证允许小型、可信验证者去独立地证明，重写的二进制代码来满足原始安全策略，从而将相对较大的二进制重写器从系统的可信计算基础中解脱出来。

一旦实施了基础框架，未来我们就可以通过结合上下文和时间策略以及数据来源策略、角色和使用方法来扩展这个框架。

32.2.4.3 强认证的组件部分

目前，Hadoop 不会对用户进行身份验证。这使得很难对安全敏感的应用程序实施访问控制，并使得恶意用户更容易回避 HDFS 实施的文件权限检查。为了解决这些问题，开源社区正在积极地将 Kerberos 协议与 Hadoop 集成（见本章参考文献［ZHAN09］）。在 Kerberos 协议之上，对于一些信息共享保障任务，可能需要添加简单的认证协议来与安全的协处理器进行认证。因此，我们向基础架构添加了一个简单的 PKI，以便用户可以使用安全的协处理器独立验证，以检索用于加密敏感数据的密钥。我们打算在实现基础设施时使用开放源码 PKI，例如，OpenCA PKI，（见本章参考文献［OPEN］）。

32.2.5 基础设施的数据组件

该组件由语义 Web 数据存储库组成。我们用 Lehigh University Benchmark（见本章参考文献［LUBM］）和 Barton 数据集（见本章参考文献［HURT06］）。LUBM 数据集是代表大学领域的综合数据集。它允许我们改变数据集中三元组的数量来测试可扩展性。Barton 数据集不是一个合成数据集。相反，它是用 MIT 图书馆目录的 RDF 表示。我们选择它主要是因为，它是评估先前研究语义 Web 的共同特征。Barton 数据集用于本章参考文献［WEIS08］中的评估。

Barton 数据集包含大量唯一的 URI：在一个有 5000 万个三元组的数据集中包含 1850 万个 URI。存储库可以建立在我们当前的集群上，也可以建立在我们使用的任何数据集的新基础架构上，具体取决于数据的大小。我们正在建立一个基础架构，可以存储高达 800TB 的数据，超过 1000 亿个 RDF 三元组。此外，由于每台机器中的磁盘空间（24TB）和主内存（24GB）都较大，因此新的基础设施可以处理更大量的数据。

32.3 将云与现有基础设施相集成

我们还将集成运行在 UTD 的 SAIAL 中的基础设施，以及在每个研究实验室中建立

的基础架构。我们还有一个额外的实验室（目前没有使用）来存储一些硬件部件。我们正在探索两种集成方法。

分离当前的基础设施和新的基础设施：在这里，我们拥有规范化形式的数据，并将当前的集群加载到新的基础设施中。之后，对基础设施上存在的规范化数据执行和测试不同的 Map Reduce 算法。优点是我们将旧硬件与新硬件分开。主要缺点是此时我们必须生成和处理大量的数据。考虑使用 LUBM 数据集的 SPARQL 查询优化模块。我们目前的基础设施可以处理总共规模为 414GB 的 55 亿个三元组。要使用我们当前的集群生成 1000 亿个三元组的大数据，我们需要进行数据生成，并将其定期转移到新的基础设施上。

将新的基础设施与当前的基础设施相结合：在这里，我们可以将当前的基础设施与新的基础设施合并，并使用组合的能力用于数据生成、数据预处理，以及执行和测试 Map Reduce 算法。Hadoop 使我们能够在任何时间点从集群中添加或删除节点。要合并它们，我们必须在同一个内部网络上设置所有的机器，这可以通过使用交换机来实现。缺点是混合不同的硬件，性能可能会受到影响。这是由于没有面向性能的变量设置。我们整个集群是具有 2 ~ 4GB 的主内存，硬盘空间大约 200GB 的几个节点（旧机器）的混合体，而其他是拥有 24GB 的主内存，磁盘空间为 8TB 的节点（新基础设施）。但是，可以通过调整 Hadoop 中可用的几个性能相关的变量来提高性能。因此，基于上述讨论，我们倾向于将我们的基础设施与当前的基础设施的训练和研究相结合。

与外部基础设施集成：UTD 团队正在使用 HP 实验室的 Open Cirrus 测试平台进行实验。而我们的基础设施将大大增强这个测试平台的能力。此外，我们计划与 NSF/DOE（美国能源部）开放科学研究人员和国防部研究人员一起，整合我们的基础设施。

32.4 利用云基础设施的示例项目

1）一种高效的隐私保护分布式数据分析综合方法。目前，我们正在为大型分布式数据集开发高效的隐私保护数据分析工具。它主要是进行一些实验的绝佳工具，以模拟使用我们的新技术挖掘大量隐私敏感电子病历。此外，这个工具使我们能够在大规模的隐私保护数据分析应用中，探索现代安全硬件（如 SCP）的可能用途。

2）用于策略规范在共享环境中实施的语义框架。作为这个项目的一部分，我们制定了一个策略框架。利用语义网络 Web 工具进行本体管理和推理。目前的语义 Web 工具面临的一个问题是，它们在存储方面和推理方面都不能很好地进行扩展。我们相信基础设施的存储、查询和推理能力，可用于实现高效的存储、查询和推理。

3）信息共享保障生命周期的框架。为了测试我们正在开发的解决方案，需要一个可扩展的平台来支持复杂的策略管理。我们的基础设施提供了这样一个平台。例如，XACML 策略的可扩展，实现可以用于跨组织的信息共享。信息共享基础设施可以增强 SCP 部分的能力，以存储、查询加密数据。这个项目让我们有机会在多个站点上分发云，并测试我们的基础架构。

4）嵌入引用监视认证。我们正在开发一套工具，用于解析、分析、执行声明指定

的 IRM 安全策略，执行策略需通过自动重写 Java 字节码二进制文件。这一努力的一个重要方面是，在大规模应用和实际的安全敏感架构中应用和测试该技术。基础设施是一个很好的平台，能用于我们的研究问题，且能针对并行高度分布式架构中实施高效的 IRM 遇到的挑战，而研发出实际的解决方案。

5）RDF 数据的关系转换。我们目前不能很好地满足对 RDF 数据的关系转换的实现（见本章参考文献［RAMA09a］、［RAMA09b］）。它可以处理有限数量的 RDF 三元组（百万级别）来执行 SPARQL 查询。然而，使用我们的基础设施对 Hadoop 的 SPARQL 查询进行优化，我们将能够处理数十亿的三元组。因此，通过利用基础设施，R2D 包装器将为语义 Web 社区提供可扩展的解决方案。

6）模式/本体匹配。为了促进本体/模式匹配（见本章参考文献［PART09］、［PART08］、［SUBB07］、［ALIP11］），我们需要提取一个高维度特征集，并确定其对应的权重。由于其序列性质，该特征提取过程对于现有技术状况是非常耗时的。我们的基础设施将能够快速提取大型功能集。因此，我们将利用我们基础架构的并行和分布式处理能力来加强我们的研究。

7）事件数据集。一个挖掘和理解异常事件的组织方法——由于汇总数据的不精确性，我们的微集群/汇总策略可能会对完整分类的准确性产生不利影响。如果我们保留原始数据以及汇总数据会更好。由于主要的内存限制，目前我们需要立即丢弃每个块的原始数据（见本章参考文献［MASU08］、［MASU09］）。然而，使用 HDFS 与 Pig Latin（见本章参考文献［GATE09］），我们将能够快速处理大量数据。这样，我们就可以在高速存储器中存储原始数据和摘要数据。此外，SSD 将促进数据的高速检索，从而大大提高分类培训和测试的速度。

8）云计算安全。我们已经开发了几种云计算安全技术，包括云数据和信息管理安全（见第五、第六、第七部分的第 13、14、15、22、23、24、25、26、27、28 章）。此外，我们还开发了云的安全模型。我们可以使用基础设施来测试技术和工具。我们将在第 34 章讨论这项研究的一些细节，并提供几个参考。

32.5 教育与实施

32.5.1 加强教育

我们提供多个信息安全和信息管理课程，包括数据和应用程序、安全、隐私、密码学、数字取证、软件和语言安全、可靠的语义 Web、数据挖掘、多媒体信息系统和网络服务。这些课程受益于基础设施。我们还正在开发云计算安全新课程，其中受到 NSF 基金的支持，并将在第 33 章中进行讨论。此外，作为数据和应用程序安全性编程项目的一部分，也是为了可信的语义 Web 课程，我们设计了课程项目，让学生们支持基础设施的开发。还有一个项目是增强 XACML 的实现，并提供基于上下文、时间和惯例的访问。因此，不仅课程从基础设施中受益，基础设施也受益于课程。

32.5.2 实施

软件组件的 SPARQL 查询处理部分是评估性能的理想选择。我们正在利用 Lehigh University Benchmark（见本章参考文献［LUBM］）和 Barton 数据集（见本章参考文献［HURT06］）两个基准来实施和评估实验。我们已经尝试了 LUBM 的数据集，从 100 万到 55 亿个三元组，该数据集提供了一个明确的 OWL 本体，并包括推理规则。许多研究项目，包括 Hexastore（见本章参考文献［WEIS08］）均已经使用 LUBM 进行评估。我们创建了一个具有 5.5 亿个三元组/数组的数据库，存储大小为 1.2TB（中间文件 – n3 格式）/ 414GB（标准化数据）。作为评估的一部分，我们正在对更大的数据集进行更多的实验。

32.6 总结和展望

在本章中，我们描述了为云安全而开发的基础设施来支持我们的项目。我们的云由硬件组件、软件组件和数据组件组成。我们正在将云与当前的基础设施相结合，并利用我们的基础设施开发了一些工具。其中一些工具在本书 4.13 ~ 4.15 节与 4.22 ~ 4.28 节与第 6 章和第 7 章中讨论过。这些工具包括云中的查询处理安全、云中的社交网络和云中的恶意软件检测。我们利用 Hadoop/Map/Reduce 技术为云提供数据存储。我们将继续加强基础设施，以满足研究和教育的需要。此外，随着技术的进步，我们将升级云。我们所面临的一个问题是建立自己的云还是利用类似 Amazon 的 EC2 等云。我们将利用我们的云和由第三方提供商提供的云。我们相信，对于我们的一些研究和教育工作，我们需要拥有自己的云。然而，为了进行大量的数据处理，我们可能会从使用类似 Amazon EC2 等云中而受益。

参 考 文 献

[AGRA06] Agrawal, R., D. Asonov, M. Kantarcioglu, and Y. Li, Sovereign joins. *Proceedings of International Conference on Data Engineering (ICDE)*, 2006.

[ALIP11] Alipanah, N., L., Khan, and B., Thuraisingham, Ontology-driven query expansion methods to facilitate federated queries, Technical report, UTDCS-30-11, The University of Texas at Dallas, 2011.

[CHU07] Chu, C.-T., S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, Map-Reduce for machine learning on multicore. *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2006.

[DEAN04] Dean, J. and S. Ghemawat, Map Reduce: Simplified data processing on large clusters Osdi'04. *Proceedings of the 6th Conference on Symposium on Operating Systems Design and Implementation*, 10 pp., 2004, p. 137–150, San Francisco, CA.

[DEVR09] DeVries, B. W., G. Gupta, K. W. Hamlen, S. Moore, and M. Sridhar, Action script bytecode verification with co-logic programming. In *Proceedings of the ACM*

SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS), June 2009, Dublin, Ireland.

[GATE09] Gates, F., O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava, Building a high-level data flow system on top of Map-Reduce: The Pig experience. *Proceedings of VLDB Endowment*, 2(2), August 2009.

[HADO] HADOOP: http://hadoop.apache.org; http://hadoop.apache.org/core/docs/r0.18.3/hdfsdesign.html

[HAMA] HAMA: http://cwiki.apache.org/labs/cloudsglossary.html

[HAML06] Hamlen, K. W., G. Morrisett, and F. B. Schneider, Certified in-lined reference monitoring on .NET. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, June 2006, Ottawa, Canada.

[HAML08] Hamlen, K. W. and M. Jones, Aspect-oriented in-lined reference monitors. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, June 2008, Tucson, AZ.

[HAML10] Hamlen, K.W., Kantarcioglu, M., and Khan, L. Security Issues for Cloud Computing, *This paper appears in the International Journal of Information Security and Privacy*, Namati, H. (ed.), 4(2), 2010, p. 36–48.

[HURT06] Hurtado, C.A., A. Poulovassilis, and P. T. Wood, A relaxed approach to RDF querying. *In Proceedings of International Semantic Web Conference*, pp. 314–328, 2006, Athens, GA.

[IBM04] IBM. IBM PCI cryptographic coprocessor, 2004. http://www.ibm.com/security/cryptocards.

[JENA] Jena: http://jena.sourceforge.net

[JONE09] Jones M. and K. W. Hamlen, Enforcing IRM security policies: Two case studies. In *Proceedings of the IEEE Intelligence and Security Informatics Conference (ISI)*, June 2009, Dallas, TX.

[LUBM] Lehigh University Benchmark (LUBM). http://swat.cse.lehigh.edu/projects/lubm.

[MAHO] Mahout: http://lucene.apache.org/mahout/

[MASU08] Masud, M., J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, A practical approach to classify evolving data streams: Training with limited amount of labeled data, In *Proceedings of IEEE International Conference on Data Mining (ICDM 2008)*, Pisa, Italy, pp. 929–934, December 2008.

[MASU09] Masud, M., J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, Integrating novel class detection with classification for concept-drifting data streams, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, Bled, Slovenia, September 7–11, 2009, 79–94, Part II.

[MCNA07] McNabb, A. W. Monson, K. Christopher, and K. D. Seppi, MRPSO: MapReduce particle swarm optimization GECCO '07: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, 177 ACM Press, 2007, London, UK.

[MORE08] Moretti, C., K. Steinhaeuser, D. Thain, and N. V. Chawla, Scaling up classifiers to cloud computers, *IEEE International Conference on Data Mining (ICDM)*, 2008, Pisa, Italy.

[MOSE05] Moses, T., ed. *eXtensible Access Control Markup Language (XACML) Version 2.0*, OASIS Standard, February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[MUYS06] Muys, A., Building an enterprise-scale database for RDF data, 2006, Netymon Technical Paper.

[NEWM08] Newman, A., J. Hunter, Y. F. Li, C. Bouton, and M. Davis, A scale-out RDF molecule store for distributed processing of biomedical data WWW '08: *Proceedings of the 17th International Conference on World Wide Web*, pp. 595–604, 2008, Beijing, China.

[OPEN] OpenCA implementation, http://www.openca.org/projects/openca/

[PART08] Partyka, J., N. Alipanah, L. Khan, B. M. Thuraisingham, and S. Shekhar, Content-based ontology matching for GIS datasets, *ACM International Symposium on Geographic Information Systems, ACM-GIS 2007*, November 7–9, 2008, Seattle, Washington, USA, pp. 407–410.

[PART09] Partyka, J., L. Khan, and B. Thuraisingham, Semantic schema matching without shared instances, *Third IEEE International Conference on Semantic Computing*, Berkeley, CA, USA, September 14–16, 2009, 297–302.

[PELL] Pellet: http://clarkparsia.com/pellet

[RAMA09a] Ramanujam, S., A. Gupta, L. Khan, S. Seida, and B. Thuraisingham, R2D: A bridge between the semantic web and relational visualization tools, to appear in *Third IEEE International Conference on Semantic Computing*, Berkeley, CA, USA, September 14–16, 2009, 303–311.

[RAMA09b] Ramanujam, S., A. Gupta, L. Khan, S. Seida, and B. M. Thuraisingham, Relationalizing RDF stores for tools reusability. *18th International Conference on World Wide Web, WWW 2009*, Madrid, Spain, April 20–24, 2009, pp. 1059–1060.

[RAMA09c] Ramanujam, S., A. Gupta, L. Khan, and S. Seida, R2D: Extracting relational structure from RDF stores, *ACM/IEEE International Conference on Web Intelligence*, September, 2009, Milan, Italy.

[ROHL07] Rohloff, K., M. Dean, I. Emmons, D. Ryder, and J. Sumner, *An Evaluation of Triple-Store Technologies for Large Data Stores on the Move to Meaningful Internet Systems 2007*. OTM 2007 Workshops, pp. 1105–1114, 2007, Part II, Vilamoura, Portugal.

[SMIT99] Smith, S.W. and S.H. Weingart, Building a high-performance, programmable secure coprocessor. *Computer Networks (Special Issue on Computer Network Security)*, (31): 831–860, 1999.

[SUBB07] Subbiah, G., A. Alam, L. Khan, and B. M. Thuraisingham, Geospatial data qualities as web services performance metrics, in *Proceedings of ACM International Symposium on Geographic Information Systems, ACM-GIS 2007*, November 7–9, 2007, Seattle, Washington, USA.

[TESW07] Teswanich, W. and S. Chittayasothorn, A transformation of RDF documents and schemas to relational databases, *IEEE Pacific Rim Conferences on Communications, Computers, and Signal Processing, 2007*, pp. 38–41, Victoria, B. C., Canada.

[WEIS08] Weiss, C., P. Karras, and A. H. Bernstein, : Sextuple indexing for semantic web data management. *In Proceedings of Very Large Data Bases (VLDB 2008)*, Auckland, New Zealand.

[ZHAN09] Zhang, K., Adding user and service-to-service authentication to Hadoop, https://issues.apache.org/jira/browse/HADOOP-4343.

第 33 章　云安全的教育项目

33.1　概述

如第 32 章所述，为了解决当前云计算平台的局限性，在 UTD 中，我们已经利用了基于 Hadoop 和 Map/Reduce 技术中最先进的硬件、软件和数据组件，以及自 2008 年以来正在为 AFRL 开发云计算安全的基础设施（见本章参考文献［AFRL］）。尤其是，我们已经使用现代硬件组件（例如，安全协处理器）来提高性能，因为要包含附加的安全功能、集成的开源软件组件，以及定制开发的软件组件，来支持对复杂数据的云查询操作安全，提供细粒度访问控制和引用监视器支持以及提供强大的身份验证机制。

要建立有效的云计算安全系统，我们还需要建立一个关于这个主题的强大的教育项目，以便我们让学生在项目上更好地工作。因此，除了在第 32 章讨论的云基础设施开发外，我们还通过 2011 年收到的 NSF 资助，建立了一套强而有力的教育项目，并提供了多项云计算安全课程。这些课程形成了一套全面的课程体系，将为其他机构的云计算安全能力建设和教育提供一个好的榜样。我们的能力建设项目，利用了我们在 UTD 中的云计算安全研究和 IA 教育方面的广泛投资，开发了云计算安全课程。特别是，我们正在开发与建设和云安全相关的新课程，同时加强网络安全、数据和应用程序安全、面向安全应用的数据挖掘、数字取证等现有的课程，这通过对每个课程引入云计算安全的主要组件来完成。我们还正在增强我们开发的现有云计算框架，以便学生可以将这个框架用于课程项目，且为框架构建功能，来将其作为课程编程项目的一部分。我们正在向与云计算研究合作的许多合作伙伴提供我们的课程。我们的云计算框架如图 33.1 所示。

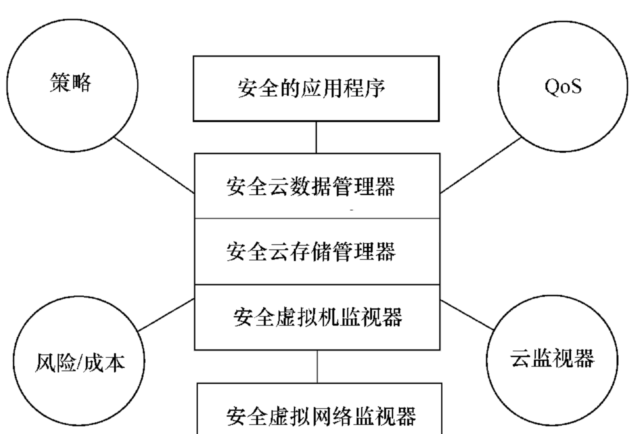


图 33.1　云计算框架

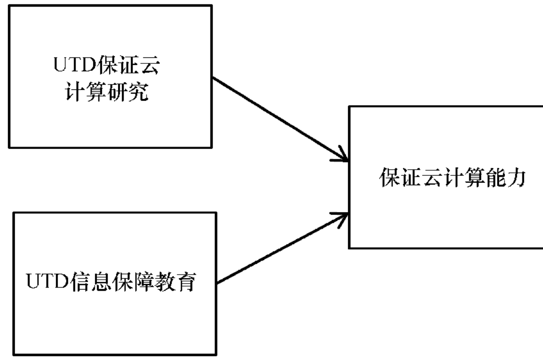


图 33.2　可确保的云计算

我们的云计算安全教育项目是建立在2000年以来UTD的信息保障（Information Assurance，IA）强大的教育项目之上的。由于2004年的教育和2008年的研究比较突出，我们的机构被指定为NSA/DHS（国家安全局/国土安全部）教育中心。2010年获得NSF SFS（奖学金）奖励，并正在培养学生获得IA硕士学位。我们的课程包括系统安全和隐私、网络安全、数据/应用程序安全、可信的Web服务/语义Web、密码学、数据挖掘安全和数字取证。如图33.2所示，我们对云计算安全的研究将与我们在IA的教育项目结合，以建立UTD的云计算安全教育的强大能力。本章的结构如下。在33.2节中，我们将描述我们目前的IA课程，包括课程、学位和证书。教育项目将在33.3节中描述。我们的评估计划将在33.4节讨论。本章总结在33.5节。本章内容如图33.3所示。我们的云计算方法的概述在本章参考文献［HAML10］中讨论。

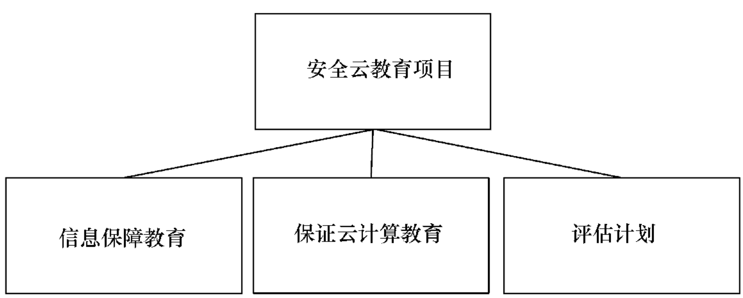


图 33.3　安全云教育项目

33.2　UTD的信息保障教育

33.2.1　UTD CS概述

就坐落在那里的高科技公司而言，UTD所处的位置仅次于硅谷。UTD计算机科学起着至关重要的作用，为高新技术产业提供毕业生和实习生以及协同支持产业研究人员。UTD CS在发展IA教育和研究的基础设施上做出了巨大投资，并明确“使国家安全”作为其6个战略目标之一。

CS部门也在发展网络安全的研究和教学基础实施上做出了重大投资。这些包括

①设立多支师资队伍，聘请计算机安全和 IA 领域的教员；②在信息安全教育（2004）和研究（2008）上努力建成 NSA 卓越中心；③成立网络安全及应急准备研究所，该机构设有网络安全研究及教育中心；④进行重大投资来建立安全分析和信息保障实验室（SAIAL），以符合 MIL－STD－285 标准；⑤在网络安全和信息保障方面开设了一些课程；⑥（自 2004 年以来）提供研究生和本科生 IA 证书、辅修 IA（2006 年以来）以及 IA（2010 年以来）的硕士文凭。

33.2.2 IA 提供的课程

CS 部门在研究生和本科阶段提供完善的课程体系。我们的课程可以在本章参考文献［CS］中找到。随着新教师的聘用，IA 课程中的新课程数量将大幅增加。目前 CS 部门的课程包括：

本科阶段

1. 计算机和网络安全
2. 数据和应用安全
3. 数字取证

硕士阶段

4. 网络安全要点（涵盖 10 个 CISSP 模块）
5. 信息安全
6. 密码学（入门级和高级）
7. 网络安全
8. 数据和应用安全
9. 数据隐私
10. 基于语言的安全
11. 建立可信赖的语义 Web 和 Web 服务
12. 安全应用的数据挖掘
13. 系统安全和取证
14. 反向工程和恶意软件分析
15. 生物识别
16. 关键的基础设施保护
17. 安全的社交网络
18. 云计算安全

作为我们云计算计划的一部分，我们引入了一些课程，包括云数据安全和安全的 WS 和云计算。这些课程将在 33.3 节中讨论。IA 部门的教育和研究工作目前正在 CS 部门的 5 个领域内进行，如图 33.4 所示。例如，计算机系统方向提供信息/系统安全课程，而理论方向提供密码学课程。我们的学生在政府实验室以及商业行业和国防建设中都能接触到 IA 的活动。例如，我们数字取证课程的学生，在实地考察中参观北得克萨斯州联邦调查局实验室，并聆听 Richardson 警察局的客座讲座。除上述 CS 课程外，

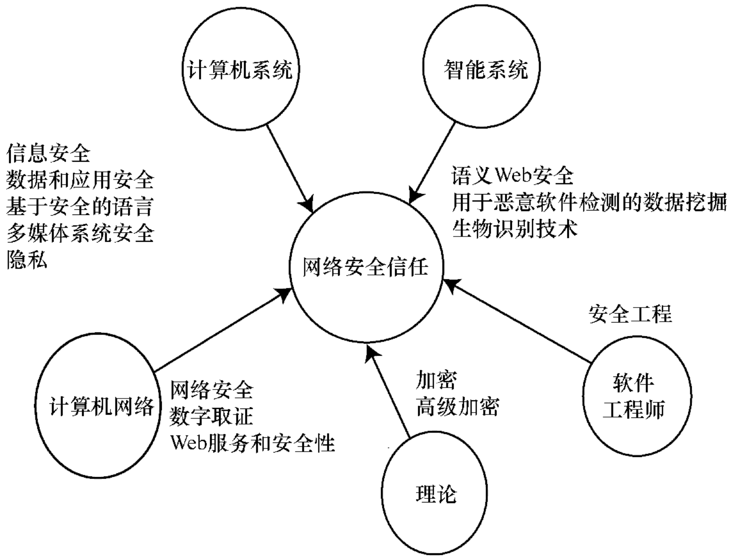


图 33.4　计算机科学课程

UTD 的管理学院（School of Management，SoM）和经济政策与政治学院（Economics、Policy and Political Sciences，EPPS）也提供相关的管理课程、IA 政策方面的课程，包括风险分析。我们正在与这些学校进行跨学科研究，并提出了一个跨学科的网络安全研究生课程。

33.2.3　我们关于 IA 的教育项目

UTD 的 CS 部门在 IA 方面设有多个课程，包括非 CS 学生的 IA 辅修课程、本科和研究生水平的认证课程，以及最近在研究生阶段开设的硕士课程。该部门还负责为 DFW（达拉斯 – 沃思堡，Dallas – Fort Worth）Metroplex 的专业人士，开设专业教育项目下的安全相关课程。本科要求学生完成 3 个本科级 IA 课程，包括数据和应用程序安全性、计算机网络安全和数字取证以及预备知识。本科证书课程向 CS 学生开放，要求学生完成上述 3 个 IA 课程。在研究生阶段，认证有多种选择。自 2003 年项目成立以来，超过 250 名学生已获得 IA 认证。CS 部门在研究生课程中提供了一个 IA 方向的课程组。在 IA 课程方向，从 2010 年秋季开始，要求学生完成一套 5 个或更多的 IA 专业课程。

33.2.4　IA 教育与研究的设备和设施

CS 部门设有多个实验室，积极用于 IA 教育和研究。SAIAL，一个最先进的实验室，成立于 2004 年，是用于研究和教育的主要设施。实验室由 3 个独立的房间组成，每个房间均经过单独测试以符合 MIL—STD – 285 TEMPEST 标准。图 33.5 给出了设施及其设备的概况。实验室的一些研究用途包括：①模拟和测试，用于识别多提供商系统和网络的安全漏洞。②利用定制和商品成品组件（Commercial – Off – The – Shelf，COTS）工具

对“安全漏洞进行基准化和量化”，“白帽”黑客技术推动了定制和 COTS 软件分析工具的使用。另外，也可以作为已知和可疑的漏洞和漏洞评估分析技术的存储库，这些漏洞包括病毒、蠕虫。③高级数字取证技术的开发与测试，以及进行取证。④在网络安全课程中进行实验练习，并尝试入侵检测和对入侵者跟踪。

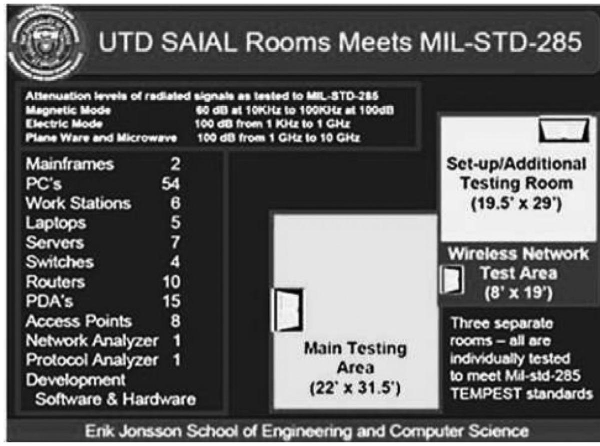


图 33.5 UTD 中的 SAIAL

除了 SAIAL，该部门还有一个计算机网络教学实验室（Computer Networks Instructional Lab，CNIL），用于我们的网络实验室课程和我们在网络安全课程中的实践实验。该实验室最初由 Jonsson 学校用得克萨斯州的资金资助而组建，包括网络设备。最近，在国防部信息保障奖学金计划（Information Assurance Scholarship Program，IASP）的能力构建补助金的帮助下，我们获得了相关设备，主要用于实际操作网络安全教育的目的。该设备包括 3 个具有多核 Opteron 处理器和 64GB 内存的高端戴尔 R805 服务器。使用 VMware 虚拟化软件系统，该设备能够支持 100 个节点的网络环境，以执行各种各样的网络安全实验室练习活动。还有 IA 教授管理的其他实验室（例如数据安全和隐私实验室、数据挖掘实验室、信息保障实验室和软件安全实验室）。此外，UTD 正在构建第 32 章中讨论的云计算安全基础架构，作为由 AFOSR 和 NSF 资助的云安全项目的一部分。

33.3 云计算教育项目的保障措施

33.3.1 能力建设活动组织

我们的能力建设项目包括两个主要部分：①课程开发和实验室开发。关于云安全的顶级项目课程是这项工作的主要成果之一。云系统是综合系统，不仅包括关键计算资源，还包含多个层次，使得它相当复杂。云安全系统涉及安全的不同方面，从数据安全到访问控制再到防御社会工程、系统级安全攻击、网络安全攻击、Rity 技术（防火墙系统，入侵检测系统等）。我们的课程强调了分布式计算的原理，如何共同构建一个安全的云。尤其是，我们使用分布式系统的已知安全漏洞中的知识，以及在保护云计算系统的背景下的已知和新的方法，去解决漏洞问题。

我们目前的云计算安全架构包括云虚拟机安全、云数据管理安全、云存储安全和云监控安全等。此外，我们正在广泛采用跨学科研究，将风险和成本分析整合到安全系统

中，以讲授基于风险的云安全管理方面的课程。我们正在密切关注云取证的其他考虑因素，并将这些方面纳入我们的课程。能力建设项目的第二部分是，建立一个具有所有必要组件的现实教学云系统，并开发必要的课程，以讲授信息保障和计算机安全的各个方面，因为它与大规模云系统的安全相关。按照 NSF 和 UTD 政策和批准程序，我们的系统将面向类似有教育目的的其他机构开放，我们也将在课程中分享课程材料和我们所使用的云教学系统。我们的课程将在 33. 3. 2 节中讨论。在 33. 3. 3 节讨论面向我们为学生编程的项目。33. 3. 4 节将讨论云计算教学设施。

33. 3. 2 课程开发活动

我们正在采取双管齐下的课程开发方式：第一个是确保云计算的课程；第二个是将组件引入到我们在 IA 中的几个关键课程。我们将在下面分别讨论两种方式的细节。我们的方法如图 33. 6 所示。

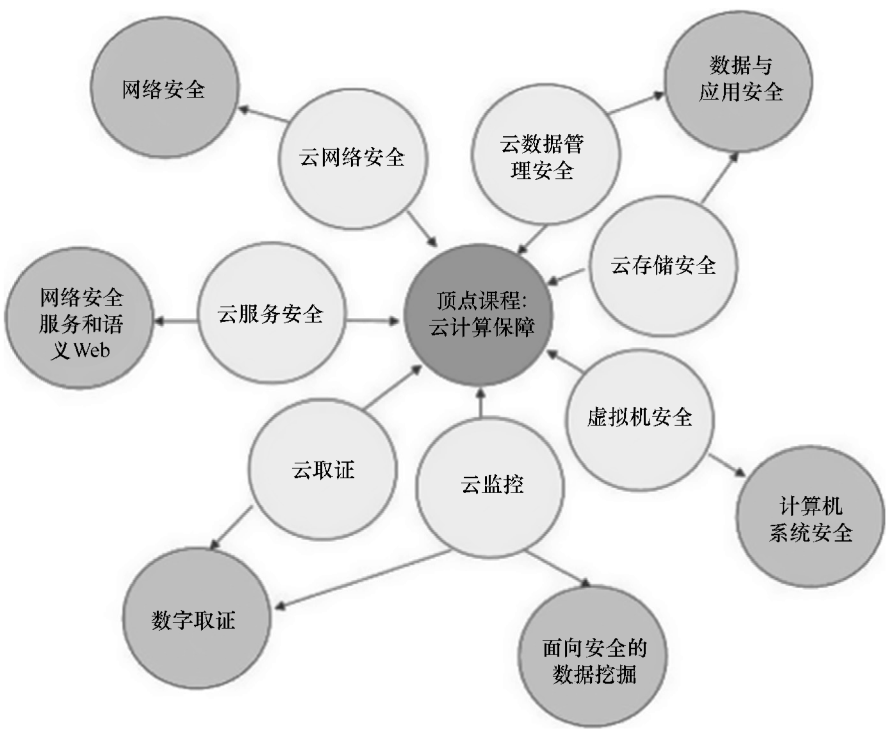


图 33. 6 云计算保障课程

33. 3. 2. 1 顶点课程

我们的顶点课程基于以下工作，云计算研究与开发；新兴的云计算保障研究、云计算原型、云计算产品和云计算标准；UTD 正在为 AFOSR 进行的研究和实验：①利用云中的虚拟化和查询处理安全实施存储管理安全；②策略管理的嵌入式参考监视器；③基于风险/激励的信息共享保障。云计算保障的顶点课程包括以下组件：①使用硬件/软件

编码的安全管理程序；②使用 VM 的云存储安全算法；③云查询处理安全算法和风险感知访问控制；④实现安全的虚拟网络监控；⑤云取证；⑥云监控工具。

云计算保障课程基于图 33.1 的框架。该框架基于分层 SOA 包括 VM 安全、云存储安全、云数据安全和虚拟网络监视器层安全。交叉服务由策略层、云监视器层、风险分析层和服务质量（Quality of Service，QoS）层提供。顶点课程的第一部分重点介绍一般云计算的原理和组件，包括管理程序、存储和数据管理以及网络组件。此外，还提供了各种标准以及 IBM、Microsoft、Oracle 和 Salesforce.com 的产品。课程的第二部分包括云计算保障的发展以及我们在这一领域的研究成果。特别地，我们的课程包括以下组件。

1）安全管理程序：作为云计算的推动者，虚拟机技术被广泛采用，它通过管理程序提供。确保管理程序的安全性对于确保云计算来说至关重要。我们的课程包括安全管理程序的开发，安全的 VMware 和 Secure XEN 平台。此外，我们正在纳入研究结果。例如，我们的研究发现，硬件/软件方法的组合对于系统保障是有效的。除了传统的执行平台，我们还在管理程序中设计和原型化组件。尤其是，我们的课程包括虚拟机中的硬件/软件组合解决方案的描述，以防范安全威胁，例如，键盘记录器、缓冲区溢出和入侵。

2）云存储管理安全：我们的课程包括与云存储系统相关的各种安全问题，如 Hadoop 框架的安全性。此外，我们也将研究结果纳入课程。例如，我们正在开发一种存储基础设施，将多个提供商的资源整合在一起，形成一个庞大的虚拟存储系统。当存储节点承载来自多个域的数据时，为每个域创建一个虚拟机以隔离信息，进行相应的数据处理。由于可以动态创建数据并将其分配给存储节点，所以有必要支持安全的 VM 管理服务，如池管理。动态创建 VM 的目的是托管数据并支持每个域的处理。我们利用线程池概念，创建了 VM 池技术。VM 池根据需求和资源限制而增长和缩小。我们正在 XEN 和 VMware 之上实施虚拟的全局云存储基础设施。

3）云数据管理安全：我们的课程包括与云数据管理相关的各种安全问题，如云查询优化和查询重写。此外，我们也在介绍研究结果。例如，我们已经使用基于 XACML 的策略管理器，并利用 Hadoop/Map/Reduce 框架（见第 23 章）为云中的 RDF 数据开发了查询处理安全算法。我们还设计了基于 Hive 框架的查询处理安全算法（见第 24 章）。这些成果以及经验正在被纳入我们的课程。另外，还包括基于我们对云计算的风险访问控制和查询处理，以及云的 QoS 的研究成果。例如，这部分还讨论了云数据处理的不同选择是如何影响安全风险的。此外，云的上下文环境也涵盖了基本的风险管理和分析工具。

4）云服务安全：我们的课程包括与云服务有关的各种安全问题。尤其是，我们研究了平台即服务、软件即服务以及基础设施即服务中各个方面的安全性。讨论了各种安全标准、产品和原型。

5）云网络安全管理：我们的课程包括与虚拟网络安全和管理相关的各种问题。云计算系统及其托管应用程序面临着基于网络的安全威胁，这种潜在的影响，使得云计算系统成为攻击的主要目标。这样的系统安全需要多层次的保障，因为潜在的安全威胁可

能来自内部和系统外部的各种实体。此外，托管的服务应用程序面临的潜在安全攻击，也可能对其他共同服务或应用程序产生负面影响。我们的课程包括对云计算系统中基于网络的安全威胁的讨论，还包括可用的检测技术和云端网络管理中减轻威胁的技术。

6）云计算安全策略管理：我们将云系统中的各种策略管理纳入到我们的课程中。此外，我们也正在将我们研究的结果纳入适用于云的嵌入式参考监视器概念中。例如，云框架通常需要更复杂的策略语言，来实现细粒度的数据机密性策略、责任策略和身份管理策略。为了支持这些策略，自定义操作系统通常是必要的。这样的操作系统在资源消耗和进程加载时间方面都会导致计算开销。自定义操作系统来支持新策略的需要，在策略语言中引入灵活性，都可能导致系统的可信计算基的负荷大量增加。为了对进程级云安全，实现更灵活、更轻量级、更高保障的保护，我们正在基于认证的嵌入参考监视器（In－lined Reference Monitor，IRM），扩展传统的管理程序架构使其具有更高级别的安全性。

7）云监控：我们正在开发两套云监控工具。第一套工具扩展了我们为云的恶意代码检测和网络流量分析而开发的数据挖掘算法。该算法用于挖掘数据流并检测新类型的恶意代码。第二套工具仅用于监视云端。例如，对于 IaaS 类型的云计算应用，我们正在开发工具来监控基础物理资源的利用率和负载分布。这些工具也正在纳入我们的课程。

8）云取证：我们的课程包含了云取证中的主题。例如，使用云进行取证分析，并确定云中的攻击来源。尤其是，包含有关如何扩展当前取证工具使其应用到云中，以及开发面向云的审计和绩效工具的信息。我们还讨论了数据提取和分析技术。虽然我们目前对云保障的研究不包括云取证，但是我们正在与普渡大学进行云计算绩效研究。例如，我们正在云中每个节点的每个作业中收集绩效数据，并进行分析。这些结果也纳入我们的课程。

33.3.2.2 组件插入现有课程

如图 33.6 所示，我们在顶点课程中，介绍的几个组件能很好地适合我们目前的 IA 课程，也指明了新组件会对当前课程有所增强。图 33.6 还说明了新组件将如何增强我们当前的 IA 课程，并为我们的顶点课程所做出的贡献。

33.3.2.2.1 计算机安全性

在本课程中，将向学生介绍信息安全方面的高级概念，特别是强调操作系统的安全性。需要注意的是，我们讨论了存储器管理、文件系统管理和进程间通信的安全。学生还可以编写病毒和蠕虫程序。我们引进了一个关于云系统安全性的附加模块，并提供虚拟机和其他管理程序安全的概述。此外，还介绍了我们对管理程序安全的研究结果。尤其是，为了防御如键盘记录器、缓冲区溢出和入侵等安全威胁，我们对虚拟机中的硬件/软件组合解决方案的描述。

33.3.2.2.2 数据和应用安全

在本课程中，我们介绍数据库中的策略管理、多层次的数据管理安全、推理问题、对象安全和多媒体系统、信息共享保障、信息集成安全、数据仓库安全、基于隐私保护的数据挖掘、社交网络安全、知识管理安全和对数据库的攻击。我们正在本课程中加入

基于云数据安全管理（如查询优化和查询重写安全，以及索引安全）的其他模块。另外，我们还介绍基于云存储安全管理的模块，如加密数据存储和基于风险的访问控制。

33.3.2.2.3　网络安全

本课程涵盖网络安全的传统主题及与互联网安全相关的更实际的主题及其应用。该课程还包括一个重要的实践训练组件，学生们在几个实际的互联网安全主题上进行练习，包括密码破解、漏洞扫描和开发。我们也正将与云中的安全挑战相关的其他问题，和解决这些问题的方案纳入到课程中。

33.3.2.2.4　数字取证

目前这是本科课程，我们计划在不久的将来，推出研究生课程。对于高年级本科课程，学生学习取证数据恢复、取证数据分析、事件重建和文件系统取证。学生还使用Encase工具进行取证分析。我们介绍一个关于云取证的新知识模块。尤其是，我们讨论如何使用云进行取证分析。我们还讨论了由于云而可能发生的新攻击，并探索扩展当前取证工具在云中运行的方法。

33.3.2.2.5　面向安全应用的数据挖掘

这是博士课程。我们将向学生介绍各种数据挖掘技术，然后探索入侵检测、恶意代码检测、缓冲区溢出检测和僵尸网络检测等网络安全应用。学生为安全应用构建数据挖掘工具。我们介绍这个新知识模块，讨论如何将云计算用于可扩展的数据挖掘。此外，我们还研究了数据挖掘是如何用于云计算的安全问题中，如审计和绩效。最近，我们推出了一个新的大数据分析课程，与面向安全应用的数据挖掘课程叠加在一起。

33.3.2.2.6　构建安全的语义Web服务

这也是博士阶段课程，我们将向学生介绍语义Web和WS技术，以及这些技术的安全问题。尤其是，对语义Web进行策略表示以及对语义Web安全技术进行了讨论，还提供了各种WS的安全标准。本课程中介绍的一个新知识模块，将讨论平台即服务和基础设施即服务的安全性。另外，我们讨论了云中语义Web数据安全处理的研究结果（如RDF数据），还讨论了如IBM、Microsoft和Oracle等公司实施的各种云安全服务。

33.3.2.2.7　密码学

在这个课程中，我们将向学生介绍各种加密协议技术。此外，还包括SMC等主题。我们正在添加使用云运行密集型加密算法的知识模块。另外，还讨论了数据存储的安全问题。

33.3.2.2.8　语言安全

在这门课程中，我们将向学生介绍语言安全概念，包括程序验证方法。本课程将纳入我们的云计算策略管理和嵌入式参考监视器模块。

33.3.3　课程编程项目

作为课程的一部分在4.2节中已经描述，学生可以进行一些编程项目，以提高他们在保证云计算方面的技能，并为我们的云计算平台做出贡献。在我们正在进行的研究的基础上，以下是我们学生的一些示例编程项目。

33.3.3.1 为了存储安全的细粒度访问控制

目前的 Hadoop 实现强制执行粗粒度的访问控制策略，实际上它把所有系统资源当做一个组，进行允许或拒绝，不区分这些资源。当数据、查询和相应的系统资源与安全性相关时，以及并不是所有用户和进程都可以完全信任时，这种粗粒度的安全性就不够了。目前的工作是通过实施 Hadoop 的标准访问控制列表，来实现对某些系统资源（如文件）的限制访问。然而，这种方法的局限性，是要执行的安全策略已经被内嵌到 OS 中，因此在不修改操作系统的情况下不能轻易地进行更改。因此，可以设计一个针对学生的课程项目，那就是在 Hadoop 上构建灵活和细粒度的访问控制策略，如 RBAC（基于角色的访问控制）、UCON（使用控制）和 ABAC（基于属性的访问控制）。

33.3.3.2 灵活认证

目前测试版本的 Hadoop 支持使用 Kerberos 和令牌进行身份验证。在 Kerberos 协议之上，为了信息保障任务，可能需要添加简单的认证协议，用来与安全协处理器进行认证。第二个编程项目是让学生去添加一个简单的公钥，使得用户可以独立地使用安全协处理器进行验证，以检索用于加密敏感数据的密钥。

33.3.3.3 虚拟机安全管理

我们已经检查了虚拟机和其他虚拟机器，并为我们的云实现并构建了 XEN，因为它是开放源码的，并且拥有优秀的文档和用户手册。XEN 还支持非 Linux 系统。此外，XEN 具有一些安全功能，可用于构建额外的安全性。第三个编程项目就是为学生在 XEN 中构建访问控制功能。

33.3.3.4 云的安全协处理器（SCP）

根本上来说，SCP 是具备有限通用计算能力的防篡改硬件。安装在服务器上时，它能够执行对服务器完全隐藏的本地计算。如果检测到篡改，则 SCP 清除内部存储器。由于 SCP 是防篡改的，可能会诱使在 SCP 上运行整个敏感数据存储服务器。将整个数据存储功能推入 SCP 是不可行的，因为：①它是防篡改外壳。SCP 通常只有有限的存储空间（只有几兆字节的随机存取存储器和几千字节的非易失性存储器）和计算能力。性能会随着时间的推移而改善，但是散热/电力使用等问题（必须进行控制以避免公开处理），将导致一般目的和安全计算之间存在差距。②另一个问题是，在 SCP 上运行的软件必须被完全信任和验证。这种安全要求意味着在 SCP 上运行的软件应尽可能简单。因此，学生的第四个编程项目是使用 SCP 为云构建安全存储。

33.3.3.5 恶意代码检测的可扩展技术

我们正在开发高效和可扩展的特征提取技术，并将这些技术应用于真正的良性和恶意可执行文件的大型语料库中。该特征提取和选择过程既是计算的又是存储密集型的。例如，在我们以前的工作中，我们从只有 3500 个可执行文件的语料库中，提取了大约 40 亿 n-grams（*n* 个连续的十六进制数字）。特征提取过程需要磁盘的 I/O 操作，因为所有功能都不能存储在主内存中。对于具有四核处理器和 12GB 内存的机器，它花费了大约 2 小时和千兆字节的磁盘空间。需要注意的是，这是静态数据集的资源需求。如果数据集是动态的，像新数据不断到达的数据流，并且该特征提取和选择将被重复执行以

发现较新的特征。这个过程将存在一个主要瓶颈。例如，我们一起考虑了 105388 个可执行文件。因此，对于这个巨大的数据集，我们以前的方法，在有限的存储和时间内可能不会很好地扩展。因此，我们正在使用我们的云计算框架开发可扩展解决方案。我们可以为学生设计编程项目，以检查我们算法中的可扩展性。

33.3.4　云计算教学设施

我们使用各种开源工具让学生熟悉基础云计算概念。对于云中的大规模数据分析，我们专注于研究云基础设施中由 Hadoop 集群组成的一部分。学生使用此集群进行编程项目。此外，学生们可以使用 Hadoop 实践上一节中描述过的各种训练实验。例如，对于数据和应用程序安全，学生可以使用这种基础设施进行编程分配，以保障信息共享。学生分为两组，每个团队都是联盟合作伙伴，并将为其他团队设计策略，也将数据存储在云中。例如，A 组为 B 组制定了一套策略，另外针对 C 组制定了一套不同的策略。当某个组（例如，B 或 C）想要访问由 A 组放置的数据时，将针对数据执行适当的策略。除结构化数据外，学生还可以尝试使用非结构化数据进行实验（例如，文本、图像）。

除了上述云，我们计划安装 Apache VCL（虚拟计算实验室）作为我们实验室环境的一部分。有关 VCL 的信息可以在本章参考文献［VCL］中找到。基本上，VCL 是一个开源系统，用于动态地为最终用户提供专门的计算机环境的远程访问。这样的系统将有助于向学生介绍基础设施作为云应用的服务类型。使用这种云计算教学设施，我们将设计各种动手实验和作业，学生可以将他们的知识应用于实际问题。我们正在创建一个完全用于教育目的的第二个云集群。通过此集群，作为顶点课程的一部分，我们将进行云端渗透和云取证演习，还可以配套进行数字取证、网络、系统和数据安全课程教学。这个云将仅用于教育目的。在项目进行期间，我们将开发详细的教学材料，其中将包括讲义和实验室练习。我们将在项目网站上进行设计和存放。此外，我们将开发用户手册。

33.4　评估计划

我们课程和实验的最好评估者是我们的学生。我们正在从学生那里获得详细的评估，并与他们讨论如何改善课程。我们还从合作伙伴那里获得资助，并利用这些资助来改进课程。我们的评估将纳入我们为 NSF 准备的中期和最终报告。在我们获得资助的基础上，我们还将为我们的云制作用户手册，以便来自各地的学生可以登录我们的系统，并学习如何使用我们的云进行教育和实验。

我们提出项目评估的三个大的阶段：项目实施效果；实现我们的目标和宗旨；项目的全面影响。第一，评估过程由两个主要部分组成。将进行形式性评估，以监测涉及的活动项目，来实施进一步完善和持续改进。这个评估阶段的目的是记录成功的经验和遇到的挑战，以及从实施阶段获得的教训，并监测项目活动状况。第二，为了实现我们的项目目标和宗旨，将进行总结性评估，评估内容包括我们的项目对学生学习产生的影响

和云计算教育的改进。

33.5 总结和展望

在本章中，我们已经建立了第 32 章中讨论的云安全基础设施，并介绍了我们的云计算教育战略。我们讨论的一些课程已经得到提供，而另外一些课程正在进行研究制订中。我们正在利用第 32 章讨论的基础设施进行课程编程项目。我们的许多学生也已成为 Hadoop/Map Reduce 框架方面的编程专家。

现在，我们正在为云中的大数据分析提供新课程，并将涉及在云中实施各种数据挖掘算法。我们的云计算安全和大数据分析课程之间存在一些重叠。我们在云计算安全方面的研究和教育在本章参考文献［CLOUD］中给出。

我们将继续加强云计算架构以及改善我们提供的课程。例如，在 2013 年春季学期（2013 年 1 月至 5 月）期间，我们提供了一个关于分析社交网络及其安全的新课程。本课程还将利用我们的云基础设施进行编程项目。具体来说，作为课程项目的一部分，我们在第 14 章讨论的 SNODOC 的工作中扩展了安全功能。也就是说，我们将开发挖掘各种社交网络数据（如 Twitter，YouTube 等）的技术，执行安全和隐私策略，并实施云中的技术（例如，使用 Storm）。我们认为，随着对大数据分析的需求，增加了对更好的云基础架构的需求，以及更多关于云计算保障的课程也会增加。在开发关于云的基础设施以及课程方面，安全性将是一个重要因素。

参考文献

[AFRL] AFOSR-Funded Initiative Creates More Secure Environment for Cloud Computing, http://www.wpafb.af.mil/news/story.asp?id=123209377

[CLOUD] Assured Cloud Computing Education, http://csrc.utdallas.edu/Education/Assured-Cloud-Computing/Home.html

[CS] Computer Science Department, the University of Texas at Dallas, http://cs.utdallas.edu/

[HAML10] Hamlen, K. W.,M. Kantarcioglu, L. Khan, B. M. Thuraisingham, Security issues for cloud computing. *IJISP*, 4(2): 36–48, 2010.

[VCL] Virtual Computing Lab, https://cwiki.apache.org/VCL/

第 34 章　云安全的研究计划

34.1　概述

正如我们在本书中讨论的那样，云计算受到了极大的关注，在保护云端方面需要做很多工作。

因此，我们开始了在 2008 年至 2014 年期间由 AFOSR 资助的合作研究项目，其中包括对云安全问题的调查。虽然我们的初步调查是网格安全，最后还是转移到了云安全。最初，与这个项目合作的两所大学分别是得克萨斯大学达拉斯分校和普渡大学。后来，我们与其他大学合作，如加利福尼亚大学欧文分校、英国皇家学院、伦敦大学、意大利英苏布里亚大学。我们还与 ADB 咨询公司在实体提取等领域进行合作，最终实现云中的算法。该项目是研究云安全问题的第一个全面的需要多组织协作的工作，包括存储、信息管理和敏感数据协作工具。该项目还着重考虑了国防部和国家安全局的需求，这些机构一直在 GIG（全球信息网格，Global Information Grid）上广泛开展工作，出现了许多具有挑战性的安全问题。在本章中，我们将介绍我们对这个项目关于云安全所做的一些进展，以及我们未来的计划。

应该指出的是我们在本书中讨论的许多实验系统都是由这个项目发展而来。在这个部分，不仅讨论我们在得州大学达拉斯分校里的研究成果，其中有些已经在前几章讨论过，而且我们也讨论我们的大学合作伙伴所做的贡献。本章的结构如下。研究成果包括开发安全的查询处理器、混合数据存储方案和云隐私，并在 34.2 节中讨论使用云进行恶意软件分析。我们的未来计划在 34.3 节中讨论。

34.2　研究贡献

34.2.1　简介

我们在云安全中的许多研究贡献，已经在本书的第六部分和第七部分进行了讨论。我们总结一些相关研究，并在本章中讨论我们的一些其他研究。我们还将讨论普渡大学根据我们在得克萨斯大学达拉斯分校领导的合作研究项目进行的研究。

我们在 3 个节次讨论该研究。在 34.2.2 节中，我们讨论云数据安全和信息管理的研究。我们将在本节中列出第六部分中讨论的具体研究原型。在 34.2.3 节中，我们讨论云安全应用的研究。也就是说，这里讨论的系统说明云提供安全服务。这些系统也在第七部分中讨论过。在 34.2.4 节中，我们将讨论云的各种安全模型。这些模型包括信息流模型、访问控制模型和绩效模型。在 34.2.5 节，我们将讨论我们正在开展的一些

研究，旨在开发安全的社交网络，最终实现云中的技术。尽管 34. 2. 2 和 34. 2. 3 节中讨论的大部分研究，都是在得州大学达拉斯分校里进行，但 34. 2. 4 节提到的研究是在普渡大学进行的。ADB 咨询公司进行的 34. 2. 5 节讨论的研究，是得克萨斯大学达拉斯分校领导的合作研究项目的一部分，该节仅提供研究的样本参考。图 34. 1 所示为研究包含的内容。

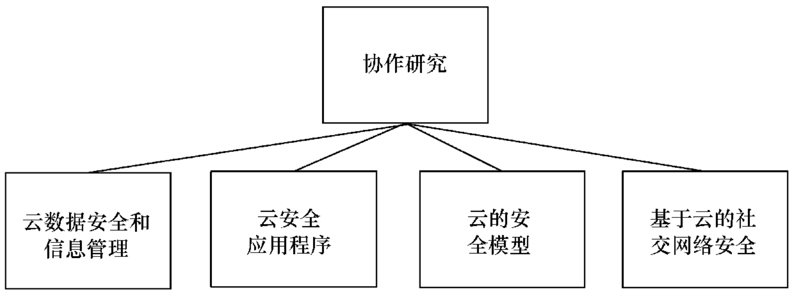


图 34. 1　协作研究

34. 2. 2　云数据安全和信息管理

34. 2. 2. 1　云中的数据密集查询处理安全

语义 Web 是通过开发各种技术来增加人类推理的新兴领域。这些技术已经被 W3C 标准化了。一个这样的标准是 RDF。随着语义 Web 技术的爆发，大型 RDF 图是常见的。这对 RDF 图的存储和检索提出了重大挑战。目前的框架并不适用于大型 RDF 图，因此不能解决这些问题。这里，我们描述了一个使用 Hadoop 构建的框架，通过利用云计算模式来存储和检索大量的 RDF 三元组。我们描述了在 Hadoop DFS 中存储 RDF 数据的方案。可能需要多个 Hadoop 作业以回答查询，因为查询中的三元组模式不能参与 Hadoop 作业中的多个连接。为了完成这个工作，我们提出了一种产生近似最优查询计划的算法，它基于贪婪方法来回答 SPARQL 协议和 RDF 查询语言（SPARQL）查询。我们使用 Hadoop 的 Map/Reduce 框架来回答这些查询。结果表明，可以在使用廉价商品类硬件构建的 Hadoop 集群中存储大型 RDF 图。此外，我们展示了这个框架是可扩展和高效的，也可以处理大量的 RDF 数据，这与传统方法不同。我们正在实施基于 XACML 的策略管理，并将其与我们的查询处理策略进行整合。我们在第 22 章讨论了系统，详细信息（见本章参考文献［KHAL10］）。

除了使用语义 Web 数据进行查询处理安全之外，我们还开发了用于关系数据的查询处理安全的原型系统。在该系统中，在 XACML 中指定策略，关系数据由 Hive 管理。我们在第 22 章讨论了这个系统（见本章参考文献［THUR10］）。

34. 2. 2. 2　混合云中的数据处理安全

云计算使得用户能够在需要时，将精确地选择计算服务成为可能。然而，用户需要根据如所需信息隐私级别、监管问题和本地计算能力等因素，来做出关于是否使用云服务的决定。鉴于这些问题，对于某些用户来说，采用混合云（公有和私有）方法，而不是仅依靠 CSP 可能是更好的选择。这样的混合解决方案，使得某些关键功能或关键

组织的任务能够在用户站点本地执行，同时允许将较不重要的任务外包给公有云。此外，这还提高了吞吐量，同时降低了高级数据安全性的运营成本。我们可以预见在采用混合云方法之前必须克服的 3 个挑战。

1）数据设计：如何分割混合云中的关系？此问题的解决方案必须考虑到关系中的敏感性以及用户的工作量。此外，解决方案必须考虑到在混合云中存储关系的资金成本，以及将敏感信息发布给公共 CSP 所带来的风险。此外，解决方案还必须能够适应用户不断变化的数据和工作负载要求。在我们的方法中，数据设计问题被形式化为多目标优化问题，其中每个目标捕获上述子问题中的一个。此外，我们已经表明这个问题是 NP 完全问题，因此我们对这个问题提供了动态规划和爬山算法的解决方法。

2）数据安全：如何通过加密保护公有云中的用户数据，同时启用对此加密数据的查询处理？这个问题的解决方案，必须确保服务提供商和入侵服务提供商的黑客都不会从存储的数据中提取任何有意义的信息。可以通过加密存储的数据来实现数据安全性。然而，加密提出了一组新的挑战，例如，加密的粒度，以及对加密数据进行查询处理。在我们的方法中，使用“桶化”技术来提供数据安全性。这种技术不会将任何数据泄露给服务提供商或黑客。此外，这种技术使我们能够有效地查询加密的数据。这使我们能够将大量的查询处理工作推送到公共 CSP。

3）查询处理：用户如何透明地对混合云发出查询请求？如何对加密和未加密的数据有效执行查询？解决这些问题，首先必须能够估算通过两个未加密和加密的数据执行查询的成本，然后这个成本估算可以用于优化混合云上的查询执行。

在我们的方法中，我们开发了一套查询重写规则，用于将给定的查询拆分成其子查询，然后可以在公有云和私有云上执行。我们使用线性成本估计模型来近似给定查询的成本。我们模型的目标是计算查询的总响应时间。该模型考虑到实际处理时间，以及将结果从公有云传输到私有云的时间。该模型利用查询重写规则，将查询拆分为公有云和私有云中的子查询。我们的方法中的查询优化是通过估计几个候选计划的成本，然后选择具有最低响应时间的计划来完成。

我们已经将上述解决方案纳入基于 Hadoop 和基于 Hive 的云计算基础架构的附加工具中，这是我们与加州大学尔湾分校联合开展的项目，我们的研究细节在本章参考文献［OKTA12］中讨论。

34.2.2.3　云中的信息集成安全

因为其成本效益和易维护等因素，像 Amazon S3 这样的云计算服务越来越受欢迎。我们已经评估了使用 S3 存储服务来存储语义 Web 数据的可行性。智慧社区的黑皮书使用几种语义数据来产生搜索结果。在我们的方法中，以安全的方式存储了 Amazon S3 上的一本 BlackBerry 数据源，从而在基于语义 Web 的框架中利用了云计算服务。在将其存储在 Amazon S3 之前，我们使用高级加密标准（Advanced Encryption Standard，AES）对数据源进行了加密。此外，我们不会将原始密钥存储在系统的任何位置。相反，密钥由两个单独的组件生成，每个都称为“密钥服务器”。然后，生成的密钥用于加密数据。我们基本上展示了如何将信息安全地集成到 Amazon 中的 WS 上。第 24 章提供了这项工作的细节。更多信息也可以在本章参考文献［PARI12］中找到。

34.2.2.4 云中社交网络的安全

我们开发了社交智能应用程序，通过从他的社交图形和他发布的消息挖掘信息，来识别用户在社交网站上的位置。我们的系统不仅可以识别用户的城市级住宅位置，而且可进一步确定用户在他的消息中可能到访过或已经谈到的特定场所或兴趣点。我们已经进行了广泛的实验，来证明该算法在准确性和运行时间方面的功效。该算法胜过所有现有的位置提取方法。为了显示算法在安全分析中的适用性，我们开发了一个强大的工具，允许分析人员识别任何 Twitter 用户和他朋友的位置，并在上下文中绑定，以便揭示世界各地的不同用户间正在谈论的内容。该工具提供了一个直观的图形接口分析器，可以使用它来可视化用户和他朋友访问的地点（由算法确定），来识别和监视潜在的安全威胁。暴动或骚乱等政治集会往往都在像 Twitter 这样的社交网站上进行组织，在目前这样的世界性场景中，我们的系统被证明是检测、识别和跟踪恶意用户的好工具。

更多关注的重点是开发信息集成以及我们系统的推理组件。特别是，来自多个社交网络的信息被安全地整合，并进行分析和推理以检测未来的事件。

我们正在为系统安全和隐私投入更多精力，并在云中实施系统。我们的工作细节在本章参考文献［ABRO09］中给出。

34.2.3 基于云的安全应用

34.2.3.1 面向演进数据流的基于云的恶意软件检测

对入侵检测的数据流分类而言至少有 3 个主要难题。第一，这些数据流通常是无限长的，导致传统的多遍学习算法不适用。第二，当攻击者做出反应并适应防御时，它们显示出强大的概念漂移。第三，对于没有任何固定功能集（如文本流）的数据流，必须执行附加的特征提取和选择任务。如果候选特征的数量太大，那么传统的特征提取技术就会失败。

针对前两个难题，我们开发了一种多分区多块集成分类器，其中使用 v – fold 分割数据从 r 个连续的数据块中训练 v 个分类器的集成，从而产生此种分类器的集成。其中使用单个数据块来训练每个分类器，与现有的多分区多块相比，这种多分区多块综合技术显著降低了分类误差。

为了解决第三个难题，新的特征提取和选择技术被开发，用于没有任何固定功能集的数据流。该技术的可扩展性通过 Hadoop/Map/Reduce 云计算架构的实现来展示。如果数据点的特征空间不固定，分类问题的一个子问题是描述每个数据点中特征的提取和选择。如在 Kolter 和 Maloof 以前的工作中，我们使用二进制 n – grams 集合作为恶意软件检测特征。然而，由于 n – grams 集合总数可能过大，我们明智地选择具有最大鉴别力的 n – grams 集合。这个选择过程正在进行的同时，随着流的推进，较新的 n – grams 集合主导较旧的 n – grams 集合。这些较新的 n – grams 集合替代了我们模型中较旧的，以便识别在特定时期的最佳特征。

对于大型数据集来说，特征提取和选择的真实性实现过程，可以说时间和存储都是密集型的（也就是时间和空间的需求量都很大）。例如，我们以前的工作从只有 3500 个可执行文件的语料库中提取了约 2.5 亿 n – grams 集合。此特征提取过程需要大量的

虚拟内存（具有相关的性能开销），因为并非所有这些特征都可以存储在主存储器中。在具有四核处理器和12GB内存的机器上，提取和选择需要大约2h的计算时间和数千兆字节的磁盘空间。这还是使用纯静态数据集，当数据集是动态时，提取和选择必须重复进行，这导致出现主要的瓶颈问题。在目前的方法中，我们考虑一个更大的105万个可执行文件的数据集，单凭我们以前的方法是不够的。

因此，我们设计了一个可扩展的特征选择和提取解决方案，其利用Dean和Ghemawat的云计算框架。根据集群节点的可用性，由于因子m，特征提取和选择的运行时间可以减少，其中m是云集群中的节点数。节点是具有廉价商品硬件的机器。因此，该解决方案也具有成本效益，因为不需要高端计算机。

关于合成数据、真正的僵尸网络流量传播和恶意可执行文件的检测，理论和经验证据都证明了，我们的方案相对于其他最新流分类技术的有效性。详细信息见第25章。更多详细信息也可以在本章参考文献［MASU11］中找到。

34.2.3.2 基于云的内部威胁检测

恶意内部人员活动的证据通常被掩盖在大数据流中。例如，累积数月或数年的系统日志。基于集成的流挖掘技术利用多种分类模型，即使在流是无限制、不断演进和未标记的情况下，也可以在流中实现高精度的异常检测。这使得该方法有效地识别了内部威胁，这些威胁随着时间变化，通过改变其行为而隐藏活动。

34.2.3.3 基于云的信息共享保障（AIS）

美国国防部对AIS的愿景是“在信息环境中具有可操纵性，通过敏捷企业提供信息的力量，以确保任务成功”。国家安全局首席信息官Lonny Anderson表示，他们也在专注于以“以云为中心”的方式与其他机构进行信息共享。为了满足美国国防部的需求，我们的项目正在开发由AFOSR资助的以云为中心的信息共享技术和工具。在我们的方法中，组织将其策略和数据放在云端，这些数据根据政策在各组织之间共享。

由欧洲航空研究与发展研究院（EOARD）资助，我们与欧洲合作伙伴（伦敦大学国王学院、意大利Insubria大学），开发了示范系统。第一个演示说明了如何根据XACML中指定的策略在云中共享信息。在第二次演示中，我们正在实施基于语义Web的策略引擎，并展示了多个社交网络可以基于语义Web策略分享我们云端的信息。对于这两个演示，我们将使用我们实施的云数据安全管理器。这些示范系统在第27章和第28章中有描述。详细信息也可以在本章参考文献［CADE12］、［THUR12］中找到。

34.2.4 云的安全模型

34.2.4.1 服务云中的一种信息流控制的细粒度模型

随着SOA日益完善和在强大的服务云的推动下，通过整合来自不同提供商的服务，快速原型设计和部署应用程序正在成为现实。虽然服务云有很多好处，但安全性仍然是一个主要问题。访问和信息流控制是服务云的安全的主要问题。我们在下面的段落中总结了我们的研究成果。详细内容请参见本章参考文献［TU10］和［SHE12］。

我们已经开发了一种支持WS组合的细粒度信息流控制的新型模型，即服务链信息流控制（Service Chain Information Flow Control，SCIFC）模型。由于在服务链中服务之

间交换的数据可能相关，为了避免不必要的信息泄露，服务链应考虑以原始或转换形式接收其数据的所有服务。在响应流中，服务链已经建立，每个服务都知道谁可以在响应中直接或间接访问信息。在这种情况下，服务 s_i 确定是否允许其响应中包含的数据直接或间接释放到服务链 s_{i-1}，…，s_1 中。在请求流中，要建立服务链，就不能确定链中的具体服务。服务 s_i 在构建下一个服务 s_{i+1} 时，需要弄清楚数据发起者是否愿意将其敏感数据从 s_{i-1} 的请求中提取出来，放到服务 s_{i+1} 中，因此，有必要支持反向检查协议，以便我们用 s 通过 s_{i-1} 检查 s_1，来决定 s_{i+1} 是否组合到链中。

执行反向检查允许链中的服务通过协作控制其敏感信息流。然而，这种机制可能导致高通信开销。为了降低协议的复杂性，我们允许服务携带并执行其他服务的安全策略。也就是说，s_1 可以将其安全策略发送到 s_2，以允许基于 s_1 和 s_2 的策略进行组合决策。

以上提供的机制有些“盲目”。也就是说，无论是远程执行反向检查还是执行安全策略，每个服务 s_i 都必须考虑所有先前服务（从 s_1 到 s_{i-1}），用于在 s_i 的请求中发送数据 Q_i。然而，如果不可能从 Q_i 中导出 s_i 的敏感数据（$1\leqslant j<i$），则当组合 s_{i+1} 时，s_i 就可以忽略 s_j。我们定义变换因子的概念来建模服务函数的变换效应，针对的是请求/响应中包含的数据。基于变换因子，可以消除信息流控制协议中的不必要的反向检查或策略执行。

有了 SCIFC 协议，服务链中的 WS 有权指定如何保护其敏感信息，而不会将关键信息泄露给不信任的服务。此外，协议不阻止任何组合形成，除非存在有不希望的信息泄露的可能性。SCIFC 协议经过精心设计，以尽量减少潜在开销，从而使细粒度信息流控制具有可行性。人们已经进行了实验来研究 SCIFC 协议的性能。结果表明 SCIFC 协议的开销平均约为 20 个服务链的 16%，考虑到其能实现细粒度安全控制的好处，这也是合理的。

34.2.4.2 CloudMask：细粒度属性访问控制

DaaS 是一种新兴的云服务，组织可以无缝地存储在云中，并根据符合法律要求和组织策略的访问控制策略进行检索。Amazon S3 和 Microsoft Azure 存储服务是目前可用的两种流行服务。在这种情况下，一个重要的要求要通过对存储在云中数据的细粒度访问控制来表示。访问控制应根据主体性质（称为身份属性）表示的策略而进行实现。电子邮件地址、用户在其组织中扮演的角色、用户访问的年龄和位置是这些身份属性的几个示例。主体应该拥有的用于访问受保护客体的身份属性称为条件。以前研究尚未解决的一个关键问题是，条件中的身份属性通常会对隐私敏感信息进行编码。

许多现有的云数据服务确实提供了一些访问控制模型。然而，在这些模型中，用户的隐私不受保护。个人和组织都认为，隐私是所有解决方案的关键要求，包括云服务，用于数字身份管理中会涉及的隐私。此外，内部威胁被认为是数据窃取和侵犯隐私的主要来源之一。通过云计算，内部威胁的范围不再局限于一个组织。

在这项工作中，我们开发了 CloudMask，一种支持基于加密的和基于属性的访问控制，同时确保访问数据中用户身份属性的隐私不被侵犯的方法。CloudMask 方法基于根据不同的对称加密密钥对数据进行细粒度加密。根据用户指定的访问控制策略，有选择地向用户提供。为了确保身份属性的可扩展性和隐私性，在 CloudMask 方法中，用户不会直接给予对称密钥。相反，用户被给予一个或多个密信，它们可以用来结合一些可变

的公共信息来即时获取密钥，它与数据一起存储在云上。CloudMask 提供向前和向后的保密，而不影响给予现有用户的密信。换句话说，CloudMask 有效地处理密钥更新操作，因为它没有直接与多个用户共享相同的密钥。密信是由基于遗忘的信封（Oblivious Commitment – Based Envelope，OCBE）技术分发给用户的。根据这项技术，只有当它们的身份属性隐藏在密码承诺中以确保隐私，在验证与数据相关的策略条件时，用户才可以解密加密的密信（即：打开信封）。这些协议已经执行了，研究的结果发表在本章参考文献［NABE10］、［DAMI10］中。

34.2.4.3　存储即服务模型中的委托访问控制

这项工作开发了一种新的方法，通过在访问控制策略改变时将数据重新加密，从而增强了 CloudMask 的性能。不同于 CloudMask 的基本方法，这里要求数据所有者在访问控制策略更改时，使用不同的密钥进行新的数据加密。另外，增强模式不需要所有者进行加密。在增强模式下，托管数据的云被委托给重新加密操作。然而，云对数据没有任何了解。

一个学者提出的方法是基于使用双层加密和增量加密。在双层加密下，第一个“粗”加密步骤由数据所有者执行，这一步使用相同的密钥加密所有数据（或大块数据）。云执行第二个加密步骤，以实现细粒度的访问控制。实验结果表明，这种方法大大降低了数据所有者的开销。这项研究在本章参考文献［NABE12］中有报告。

34.2.4.4　基于属性的组密钥管理方案

基于属性的系统可以在一组用户中进行细粒度的访问控制，每个用户由一组属性标识。安全的协作应用程序，如云启用的应用程序，需要这种基于属性的、灵活的系统来管理和分发组密钥。然而，目前的组密钥管理方案设计不适合根据组成员的属性来管理组密钥。

在这项工作中，我们定义和实施了新的密钥管理方案，允许其属性满足特定访问控制策略的用户导出组密钥。由于组成员的连接或离开而使组改变时，我们的方案有效地支持密钥更新操作。在密钥更新操作期间，发给现有成员的私人信息不受影响，只有公共信息被更新才能更改组密钥。我们的方案基于表达式，它们能够通过一系列属性来支持任何单调的访问控制策略。我们的方案是抵制串通攻击的，组成员无法集合其属性，并导出不能单独导出的组密钥。实验结果表明，我们的基础设施高效实用，这比基于属性加密（ABE）的方案效率更高。这项研究报告在本章参考文献［NABE11］中。

34.2.4.5　云中的隐私保护访问控制

存储在云中的数据往往是对敏感信息编码，并应受到各种组织策略和法律规定的约束。解决安全和隐私的常用方法，是将数据上传到云端之前进行加密。单独的加密是不够的，因为组织必须对数据执行细粒度的访问控制。这种控制通常基于如组织中的数据用户角色、用户正在进行的项目等信息。因此，一个重要的需求是，基于表达式访问控制语言指定策略，对云中托管的加密数据实施细粒度访问控制。特别地，如 XACML 的表达式访问控制模型，允许根据客体的属性（被称为身份属性），来指定对受保护客体的访问控制策略。电子邮件地址、用户在组织中扮演的角色、用户访问的年龄和位置是这种身份属性的几个示例。主体应该拥有的用于访问受保护客体的身份属性称为条件。

这种基于属性的访问控制模型对于支持数据的细粒度访问控制策略至关重要。

这方面的一个关键问题是访问控制策略条件中的身份属性，通常含有隐私、敏感信息的编码。因此，如果用户的身份属性未被保护，则就不会保护用户的隐私。个人和组织都认为，隐私是所有解决方案的关键要求，包括云服务，用于数字身份管理中会涉及隐私。此外，内部威胁被认为是数据窃取和侵犯隐私的主要来源之一。通过云计算，内部威胁的范围不再局限于一个组织。因此，在组织和云中实施基于属性的访问控制时，保护用户的身份属性是一个重要的要求。这项研究的结果在本章参考文献［BERT09］、［NABE12］中有报告。

34.2.4.5.1　网格中的绩效

网格计算系统提供大量的计算资源。例如，计算能力、数据存储和网络带宽。网格绩效使得可以审核用户和资源提供者的活动，并通过收集和分析绩效数据来调查安全漏洞。然而，绩效数据难以获得，因为网格的复杂性和异构性以及缺乏合适的绩效制度。我们以前的工作已经开发出一种解决绩效计算的方法，也就是基于一些特殊目的的代理(被称为绩效代理)。这些代理被放置在跨网格的战略位置，并根据一些政策（称为绩效政策）收集绩效资料。绩效策略规定了收集哪些资料和何时收集这些资料。更重要的是，如何协调不同管理域之间的数据收集。

在这种情况下的一个主要问题是出现了各种网格节点之间的冲突，因为并非所有节点都可以（或愿意）收集所需的绩效数据。为了解决这个问题，以实现灵活的绩效系统，我们制定了一个基于配置文件的策略选择机制。在这种方法下，最佳绩效策略的选择，要基于作业的属性和网格节点，以及每个节点收集绩效数据的能力。选定的策略保留最低层次的绩效，并且近似共享策略的要求。本研究的结果在本章参考文献［LEE11a］和［LEE11b］中给出。

34.2.5　在云中建立安全的社交网络

34.2.5.1　安全的社交网络

自20世纪90年代中期以来，社交网络为用户提供了互动的方式，反映了社会网络或社会关系。例如，谁分享兴趣或活动。社交网站新兴的前沿趋势是“实时”和“基于位置”的概念。那么什么使基于位置的社交媒体服务如此重要？

隐私和安全：发布基于位置的社交网站的更新，并将您当前的位置发布给用户，可能会导致垃圾邮件发送者的个人攻击等问题，对您的安全构成威胁。

用户位置的可信度：在某些情况下，如2009年伊朗选举的政治局面，监控数据的组织能够核实用户的位置变得非常重要。

广告营销：社交网络以较低的成本使人们相互联系，对于希望扩大联系的企业家和小企业来说可能是有益的基地。这些网络经常作为公司销售产品和服务的客户关系管理工具。

用户在社交媒体中强调位置的重要性，重要的是要理解，由于许多原因用户不明确提供这个信息。一些用户关心他们的隐私和安全性，其他人不能发现他分享地点的任何动机。除了不介绍其位置的这类用户外，还有其他人提供不正确或不可读取的位置，或

仅显示州/国家。由网络俚语和不完整句子组成的文本中的非结构化和自由形式，使用传统的自然语言处理和基于地名词典的数据挖掘方法产生不准确的结果。

我们已经开发了社会智能系统并整合了多个社交网络，确定用户的位置和许多其他属性，并分析综合信息及预测威胁。我们的研究也影响了 SNODSOC 等系统。一些细节可以在本章参考文献［ABRO09］、［ABRO10］、［ABRO12a］、［ABRO12b］中找到。

34.2.5.2 数据的可信度

普渡大学与我们合作解决了社交网络的数据可信度问题。这项工作已经在匿名社交网络的存储库存在的情况下，研究了社交网络中数据可信度的问题，并且设计和验证了一种可以在社交网络中，为用户资料（或用户简介中的特定信息）分配信任分数的方法。信任分数是从 0 到 1 的数字指标，这表明对相关信息是真实的置信度。接近 0 的值表示低置信度，而接近 1 的值表示高置信度。需要注意的是，这个信任分数只是一个指标，关于某一信息是否可信赖的最终决定，可能需要额外的分析步骤。我们开发的方法是基于将感兴趣的社交网络中的信息与来自其他社交网络的匿名数据进行比较（称为参考社交网络）。详情请见本章参考文献［LIM12］。

34.2.5.3 文本挖掘与分析

在已故詹姆斯·约翰逊博士和安妮塔·米勒女士的领导下，ADB 咨询公司与我们合作，探索了语义处理方法，以确定子句级别的文档之间的相关信息。随着研究的进展发现语义图匹配方法产生了与两者相关的有意义的信息，以及链接到与参考图匹配的图形中新增加的信息。还发现可以使用相关性和增强信息措施对相关信息进行排序。显而易见的是，开发的方法适用于其他领域，如情报、新闻报道监控、识别网络威胁和攻击，以及利益驱动的互联网搜索信息。测试方法是交换匿名活动家和联邦调查局公布的报告中的网络威胁信息，其结果令人鼓舞。

这项研究大大扩展了自由文本的语义分析：量化语义内容和语义上下文；纳入 DLSafe规则和引渡假设，模拟过程并产生推论，增加匹配相关内容的可能性并发现新知识；创建一个新的扩展的语义图结构的严格定义，其语义相关性度量用于量化以前未达到的级别的识别信息。这些新技术为跨域应用打下了基础，包括对支持国家情报分析师从大量自由文本中确定重点信息。研究的一些细节可以在本章参考文献［JOHN12a］、［JOHN12b］和［JOHN12c］中找到。

34.3 总结和展望

在本章中，我们概述了关于云保障的研究。这是关于数据和应用安全更大举措中的一部分，还包括数据来源安全和安全应用的语义 Web 技术等主题。本章重点介绍了我们 4 个方面的贡献，它们是云数据安全和信息管理、使用云的安全应用、为云和技术开发的安全模型，以及开发云中的安全社交网络。

我们开发的技术与我们定义的云安全框架的各个层次相关。我们认为没有办法开发一个安全的云。也就是说，我们需要混合和匹配一系列技术来满足我们的应用需求。挑战在于以安全的方式集成多种安全技术来开发安全的云。正如我们之前所说的，几乎不

可能提供100%的安全。因此，我们需要设计云安全，将尽可能为执行的任务提供安全性。也就是说，挑战是从可能受到损害的组件中开发安全的云。本章参考文献［UTD10］给出了从不可信组件中开发安全系统的一些挑战。我们需要开发类似的解决方案来保护云。

参考文献

[ABRO09] Abrol, S. and L. Khan, Tweethood: Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining. *SocialCom/PASSAT*153–160, 2010, Minneapolis, MN.

[ABRO10] Abrol, S. and L. Khan, TWinner: Understanding news queries with geo-content using Twitter. *Geographic Information Retrieval (GIR)*, 2010, Zurich, Switzerland.

[ABRO12a] Abrol, S., L. Khan, V. Khadilkar B. M. Thuraisingham, and T. Cadenhead, Design and implementation of SNODSOC: Novel class detection for social network analysis. *Intelligence and Security Informatics (ISI),* 215–220, 2012, Washington, DC.

[ABRO12b] Abrol, S., L. Khan, and B. M. Thuraisingham, Tweecalization: Efficient and intelligent location mining in Twitter using semi-supervised learning. *Collaborate Com* 2012:514–523.

[BERT09] Bertino, E., F. Paci, and R. Ferrini, N. Shang Privacy-preserving digital identity management for cloud computing. *IEEE Data Eng. Bull.* 32(1):21–27, 2009.

[CADE12] Cadenhead, T., M. Kantarcioglu, V. Khadilkar, and B. M. Thuraisingham, Design and implementation of a cloud-based assured information sharing system. *MMM-ACNS* 2012:36–50, St. Petersburg, Russia.

[DAMI10] Damiani, M. L., C. Silvestri, and E. Bertino, Analyzing semantic locations cloaking techniques in a probabilistic grid-based map, *Geographic Information Systems* (*GIS)*, 522–523, 2010, San Jose, CA.

[JOHN12a] Johnson, J., A. Miller, L. Khan, and B. Thuraisingham, Extracting semantic information structures from free text law enforcement data. *IEEE Intelligence and Security Informatics (ISI)*, Washington, DC, July 11–14, 2012.

[JOHN12b] Johnson, J., A. Miller, L. Khan, and B. Thuraisingham, Measuring relatedness and augmentation of information of interest within free text law enforcement documents. *2012 IEEE European International Security and Informatics, (EISI)*, Odense, Denmark, Aug 22–24, 2012.

[JOHN12c] Johnson, J., A. Miller, L. Khan, and B. Thuraisingham, Graphical representation of semantic information, *International Conference on Semantic Computing (ICSC),* Palermo, September 19–21, 2012.

[KHAL10] Khaled, A., M. Husain, L. R. Khan, K. Hamlen, and B. M. Thuraisingham, A token-based access control system for RDF data in the clouds, In *Proceedings of 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2010)*, Indianapolis December 2010.

[LEE11a] Lee, W., A. Squicciarini, and E. Bertino, Profile-based accountability policies in grid computing systems. *Proceedings of IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*, June 2011, Pisa, Italy.

[LEE11b] Lee, W., A. Squicciarini, and E. Bertino, Detection and protection against distributed denial of services in accountable grid systems. *Proceedings of 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2011, Newport Beach, CA.

[LIM12] Lim, H.-S., G. Ghinita, E. Bertino, and M. Kantarcioglu, A game-theoretic approach for high-assurance of data trustworthiness in sensor networks. *International Conference on Data Engineering (ICDE)*, 2012:1192–1203, Arlington, VA.

[MASU11] Masud, M. M., T. Al-Khateeb, K. W. Hamlen, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, Cloud-based malware detection for evolving data streams. *ACM Trans. Manage. Inf. Syst.* 2(3):16, 2011.

[NABE10] Nabeel, M., N. Shang, J. Zage, and E. Bertino, Mask: A system for privacy-preserving policy-based access to published content. *SIGMOD Conference*:1239–1242, 2010, Indianapolis, IN.

[NABE11] Nabeel M. and E. Bertino, POSTER: Towards attribute based group key management. *ACM Conference on Computer and Communications Security:* 821–824, 2011, Chicago, IL.

[NABE12] Nabeel, M. and E. Bertino, Privacy-preserving fine-grained access control in public clouds. *IEEE Data Eng. Bull.* 35(4):21–30, 2012.

[OKTA12] Oktay, K. Y., V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra, and B. M. Thuraisingham, Risk-aware workload distribution in hybrid clouds. *IEEE CLOUD*:229–236, 2012, Honolulu, HI.

[PARV12] Parveen, P., N. McDaniel, V. S. Hariharan, B. M. Thuraisingham, and L. Khan, Unsupervised ensemble based learning for insider threat detection. *SocialCom/PASSAT*:718–727, 2012, Amsterdam, The Netherlands.

[PARI12] Parikh, P., M. Kantarcioglu, V. Khadilkar B. M. Thuraisingham, and L. Khan, Secure information integration with a semantic web-based framework. *IRI*:659–663, 2012, Las Vegas, NV.

[SHE12] She, W., I. Yen, B. Thuraisingham, and E. Bertino, Security-aware service composition with fine-grained information flow control to appear. *IEEE Transactions on Services Computing.*

[THUR10] Thuraisingham, B. M., V. Khadilkar, A. Gupta, M. Kantarcioglu, and L. Khan, Secure data storage and retrieval in the cloud. *Collaborate Com:* 1–8, 2010, Chicago, IL.

[THUR12] Thuraisingham, B. M., V. Khadilkar, J. Rachapalli, T. Cadenhead, M. Kantarcioglu, K. W. Hamlen, L. Khan, and M. F. Husain, Cloud-centric assured information sharing. *PAISI*:1–26, 2012, Kuala Lumpur, Malaysia.

[TU10] Tu, M., P. Li, I.-L. Yen, B. Thuraisingham, and L. Khan, Secure data objects replication in data grids. *IEEE Transactions on Dependable and Secure Computing* , 7(1): 50–64, January–March 2010.

[UTD10] Securing the Execution Environment Applications and Data from Multi-Trusted Components, Technical Report, UTDCS 3–10, The University of Texas at Dallas, 2010.

第 35 章　本书总结和展望

35.1　关于本章

这一章为本书进行收尾。我们讨论了几个方面，包括服务安全技术、语义 Web 安全技术、云计算技术、云计算安全技术、云计算实验系统、云计算安全实验系统、用于安全应用的云计算系统、构建可信云和开发基础设施、教育倡议和云安全研究计划。讨论的主题包括云虚拟化、云数据管理、云存储以及云功能的安全问题。另外我们还讨论了新兴的商业产品。实验系统是我们在 UTD 开发的系统，包括云查询安全处理系统以及基于云的信息共享保障系统。

本章的结构如下。在 35.2 节，我们将总结本书。该总结选自每章的总结部分。在 35.3 节中，我们讨论云计算安全的方向。在 35.4 节中，将讨论我们在云计算安全中的目标。在 35.5 节中，在现有技术基础上，我们将给出未来发展方向的建议。

35.2　本书总结

我们总结每一章的内容，它们基本上均摘自每章的总结和展望部分。第 1 章简单概述了这本书。我们首先简要介绍了云计算的支撑技术，包括信息安全以及数据、信息和知识管理。然后，我们介绍了本书中讨论的各种主题，包括 Web 服务安全和语义 Web 安全，这些都是云计算安全的核心。我们还讨论了云计算和云计算安全。我们的框架是一个九层框架，每层都在本书的对应部分中进行了阐述。该框架如图 1.11 所示。我们在图 35.1 中重复了这个框架。

这本书分为九部分。第一部分描述了安全和数据管理技术，由 3 章组成：第 2、3 和 4 章。第 2 章概述了计算系统的演进。第 3 章概述了安全系统。第 4 章讨论了数据、信息和知识管理。

描述服务安全的第二部分包括 3 章：第 5、6 和 7 章。第 5 章讨论了 Web 服务和 Web 服务安全。特别是，我们首先讨论了服务的含义。接下来，我们讨论了面向服务的计算中的高级概念。然后讨论了通过面向服务的档案化和 Web 服务实现面向服务的信息系统。我们还讨论了服务的安全问题。第 6 章讨论了语义 Web 服务，因为我们的几个云原型使用了语义 Web 技术。专用的安全 Web 服务和安全性是第 7 章的主题。

描述云计算的第三部分包括 5 章：第 8、9、10、11 和 12 章。我们在第 8 章中概述了云计算。第 9 章讨论了云计算功能。第 10 章讨论了云数据管理，这是我们研究与开发的重点。第 11 章讨论了移动云。云计算产品和标准是第 12 章的主题，最后是一些专

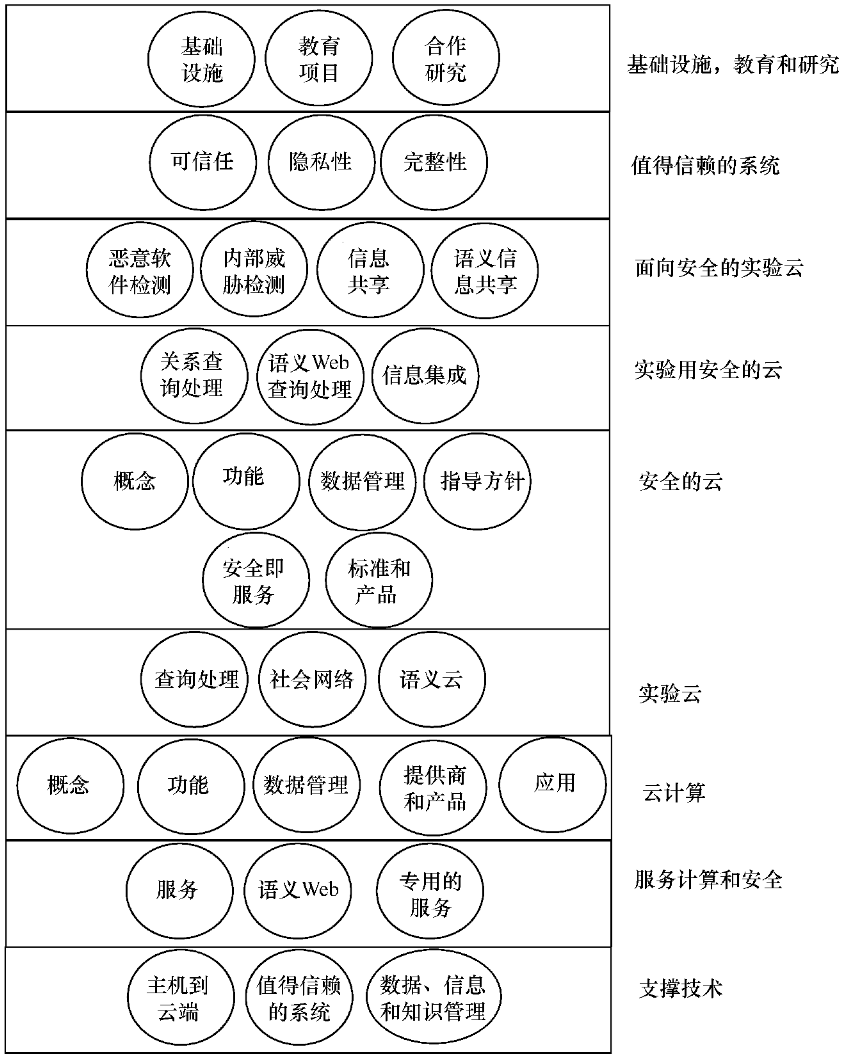


图 35.1 用于构建和保护云的分层框架

门的云服务。

第四部分描述了云计算实验系统，包括 3 章：第 13、14 和 15 章。第 13 章讨论了基于语义 Web 云查询处理的原型系统。第 14 章讨论了我们在云上托管社交网络的工作。第 15 章讨论了我们开发的多种原型，包括用于社交网络的云计算、用于语义 Web 数据存储的云计算和用于基于本体的查询处理的云计算。

第五部分描述云计算安全，由 6 章组成：第 16、17、18、19、20 和 21 章。第 16 章概述了云计算安全的概念。第 17 章讨论了云计算安全功能。云数据安全管理是第 18 章

的主题。第 19 章讨论了云计算安全指南。第 20 章讨论了安全即服务的概念。第 21 章讨论了云计算安全产品。

第六部分描述了云计算安全实验系统，包括 3 章：第 22、23 和 24 章。我们在云中的关系数据查询安全处理器已在第 22 章中讨论。我们在云中的语义 Web 数据查询安全处理器已在第 23 章中讨论。我们在云中信息集成安全工作已在第 24 章讨论。

第七部分描述了面向云的安全应用，其中包括 4 章：第 25、26、27 和 28 章。我们在第 25 章讨论了我们基于云的恶意软件的检测系统。我们在第 26 章讨论了云中的内部威胁检测。第 27 章讨论了我们在云中的信息共享保障系统。第 28 章讨论了我们在语义云中保障信息共享的方法。

第八部分由 3 章组成：第 29、30 和 31 章。重点讨论了我们关于开发可信赖云的想法。在第 29 章中，我们讨论了云服务的信任管理。在第 30 章中，我们讨论了云服务的隐私问题。在第 31 章中，我们讨论了云服务的完整性和数据质量。

第九部分由 4 章组成：第 32、33、34、35 章。它讨论了我们建立基础设施、教育项目和云计算安全研究项目的方法。第 32 章讨论了我们开发的云计算安全基础设施。在第 33 章中讨论我们在云计算安全中的教育计划。最后，在第 34 章中，我们讨论了我们关于云计算安全的多组织协作研究计划。第 32、33 和 34 章的内容构成了我们在云计算安全中的实验工作的基础。第 35 章对本书进行了总结。

这本书有 4 个附录。在附录 A 中，提供我们所写的所有书籍是如何相互关联的信息。在附录 B 中，我们讨论数据挖掘技术。我们在云计算安全中的研究，已经利用了许多这些技术。在附录 C 中，我们讨论数据安全管理。我们在云数据安全管理方面的工作所基于的概念在附录 C 中讨论。最后在附录 D 中，我们讨论我们对保障信息共享的合作研究，这为我们基于云中信息共享保障工作提供了主要动力。

正如我们所强调的，在该领域有许多发展，我们不可能列出所有成果。我们提供了一个广泛但相当全面的本领域的概况。本书面向技术管理者以及想要了解这一领域的技术人员。它也适用于希望在数据和应用安全方面（特别是一般的云计算安全）研究的学生。

35.3 云计算和云计算安全的方向

云计算和云计算安全有许多方向。我们在本书所讨论的每个主题中会论述其中的一些。图 35.2 说明了方向和挑战。

35.3.1 安全的服务

Web 服务和 SOA 是云的核心。虽然 Web 服务有许多发展，但是对 Web 服务安全仍需要进行大量研究。此外，全球信息网格和以网络为中心的企业服务等重大设施，都是基于网络服务和 SOA。因此，通过使用语义 Web 来确保这些技术以及使 Web 服务更加智能化，这对于下一代云来说至关重要。

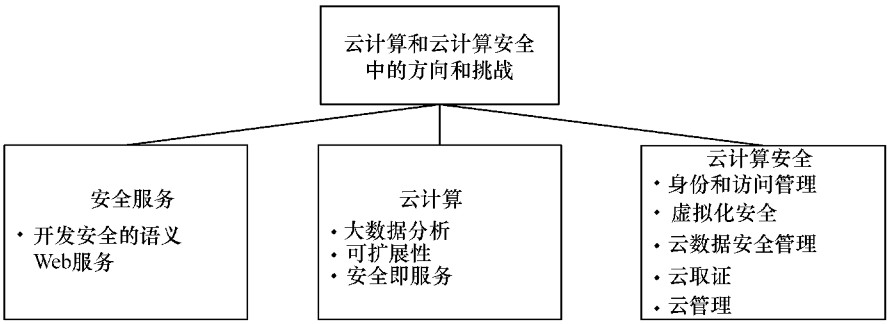


图 35.2　云计算和云计算安全中的方向和挑战

35.3.2　云计算

最近有很多关于云计算的工作，包括虚拟化、云数据存储和云数据管理。云计算的主要挑战是处理大量的数据和过程。也就是说，使用云解决大数据问题将是一个挑战。此外，为了安全应用，还应该审查云计算。例如，恶意软件检测和电子邮件中的垃圾邮件过滤。

35.3.3　云计算安全

我们要检查云中的虚拟化、存储、数据管理和网络的安全性。关于云中的身份管理和访问控制，我们还有很多工作要做。我们需要适合云的安全模型。例如，需要检查如 ABAC 之类的模型。我们需要开发类似于 SAML 和 XACML 的标准，以包含更复杂的细粒度访问控制形式。随着 Web 服务爆炸式的发展，我们在云上进行越来越多的交易，并参与社交网络，因此保护个人身份以及确保授权访问至关重要。虚拟化安全以及云取证和审计也是需要大量工作的领域。最后，我们需要确定的方面类似于管理、业务连续性规划、法律方面以及云的运营管理。

35.4　我们保护云的目标

虽然我们已经讨论了许多关于云计算和云计算安全的概念，如虚拟化、云取证和管理方面，但我们的大部分研究仍集中在云数据安全管理上。因此，我们的主要目标之一是继续开发云数据安全存储系统、云数据安全管理系统以及实施恶意软件检测、内部威胁检测以及云中信息共享保障等应用。

我们研究的一些主要方面是提取来自结构化和非结构化的众多数据源的实体，及实体之间关系。从提取的实体和实体之间的关系构建网络。分析网络，并从网络中提取对分析人员有用的信息。我们的目标是实现在云中开发的社交网络系统。此外，随着我们在社交网络的安全和隐私方面取得的进展，我们还将在云中实施安全的社交网络。

我们在 UTD 的团队正在进行云虚拟化安全、云取证和形式策略分析的广泛研究。因此，我们将继续探索云计算的基本问题。图 35.3 说明了我们保护云的目标。

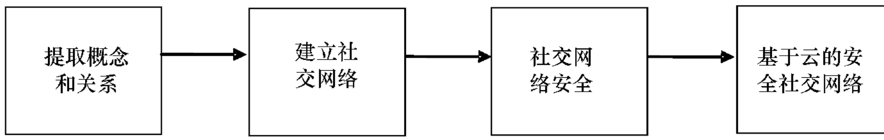


图 35.3　我们保护云的目标

35.5　我们该怎么继续发展

本书着重于服务安全、云计算和云计算安全。在 35.3 节中，我们已经在这一领域提出了许多挑战。如果要在这个非常重要的领域取得进展，就要继续研究开发。

问题是我们该怎么发展？首先，希望在这一领域工作的人员必须熟练掌握支撑技术，包括服务、语义 Web、信息安全和数据、信息管理和知识管理。例如，了解构成 Web 服务的技术以及它们如何用于云端是很重要的。此外，还需要了解正在开发的许多标准，并且能够确定最适合其组织的标准。接下来，因为该领域正在迅速扩张，而且此领域也在不断发展。读者必须跟上发展，包括阅读商业产品和原型。最后，我们鼓励读者对产品进行实验，并开发安全工具。这是熟悉特定领域的最佳方式。在实践中解决问题，并提供解决方案来更好地理解。开发人员应该熟悉 Hadoop、Map/Reduce、HBase 和 Storm 等技术。云将继续对处理和存储大量数据产生重大影响，因此云的安全性至关重要。这样，组织将更加舒适地将其数据存储在云端。

我们需要联邦和地方政府等资助机构的研发支持。美国国家科学基金会，美国国家安全局，美国陆军、海军、空军，美国国防高级研究计划局、美国情报高级研究计划局和美国国土安全部等机构正在资助安全研究。空军专注于云安全。我们还需要商业公司投资研发资金，以便在工业研究方面取得进展，并能够将研究成果转化到商业产品中。我们还需要与国际研究界合作，以解决问题从而提出不仅符合国家利益，也符合国际利益的标准。综上所述，我们需要公共、私人、学术的合作伙伴，以在云安全的重要领域上开发突破性技术。

第九部分总结

第九部分的章节内容描述了基础设施、教育项目，以及我们正在 UTD 研发的研究计划。这项工作在前面部分讨论过的原型中发挥了重要作用。

在第 32 章，我们描述了为云安全开发的基础设施，以支持我们的项目。我们的云由硬件组件、软件组件和数据组件组成。我们正在将云与当前的基础设施相结合，并开发了一些利用我们基础设施而完善的工具。这些工具已经在本书的某几个章节中讨论过，包括云中的查询处理安全、云中的社交网络和云中的恶意软件检测。我们利用 Hadoop/Map/Reduce 技术为云提供数据存储功能。

在第 33 章，我们讨论了云计算安全教育项目。特别是在云计算安全方面，我们开发了多个课程，包括云数据安全管理、云数据安全存储、并确保云取证。此外，我们还开发了云计算安全的旗舰课程。

在第 34 章，我们讨论了 UTD 主持的关于云计算安全的各种研究项目。特别是，我们讨论了云查询处理安全、云安全模型，以及在云中构建安全的社交网络的工作。

附　　录

附录 A　数据管理系统——发展与趋势

A.1　概述

附录 A 的主要目的是讲述我们以数据管理、数据挖掘和数据安全方面为背景而编写的一系列书籍。我们的这一系列书籍始于 1997 年的《数据管理系统的演进与互操作》(见本附录参考文献［THUR97］)。我们随后的书是从这第一本书演变的。我们在后续书籍的附录 A 中，基本上重复了第一本书的第 1 章内容。附录 A 的目的是提出数据管理系统的概述。然后将讨论我们所写的书籍之间的相互关系。

正如我们的系列书籍中所述，信息系统技术的发展，已经促进了许多业务领域中众多应用的计算机化。数据已经成为许多组织的关键资源，因此，高效地获取数据、共享数据，以及从数据中提取信息，并利用信息已经成为迫切的需求。因此，我们已经对分散在多个站点中的各种数据源进行整合而做出了一些努力。这些数据源可以是由数据库管理系统所管理的数据库，也可以是简单的文件。为了支持多个数据源和系统之间的交互操作，我们正在开发各种工具。这些工具使一个系统的用户以高效和透明的方式访问其他系统。

我们将数据管理系统定义为管理数据的系统，从数据中提取有意义的信息，并加以利用。因此，数据管理系统包括数据库系统、数据仓库和数据挖掘系统。数据可以是关系数据库中的结构化数据，也可以是非结构化数据。例如，文本、语音、图像和视频。过去有很多关于区分数据、信息和知识的讨论。我们并没有试图要明确这些定义。对于我们而言，数据可能只是位和字节，或者它可以给用户传达一些有意义的信息。然而，我们将区分数据库系统和数据库管理系统。数据库管理系统是管理包含持久性数据的数据库组件。数据库系统由数据库和数据库管理系统组成。

数据管理系统发展与互操作的一个关键部分体现为异构数据库系统的互操作性。自 20 世纪 70 年代末以来，已经报道了数据库系统之间互操作性的一些成就。然而，只是最近才发现在异构数据库系统中的商业发展。主要的数据库系统提供商都在提供其产品和其他系统之间的互操作性。此外，许多数据库系统提供商也正在向称为客户端 — 服务器的架构中进行迁移，这有利于增强分布式数据管理的能力。除了需要在不同的数据库系统和客户端 — 服务器环境之间的互操作性的努力之外，其他工作还涉及处理自治和联合环境。

附录 A 的结构如下。由于数据库系统是数据管理系统的关键组成部分，我们首先

概述数据库系统的发展情况。这些发展在 A.2 节中讨论。然后我们在 A.3 节中提出对数据管理系统的愿景。我们的数据管理系统框架将在 A.4 节中讨论。需要注意的是，数据挖掘、仓储以及 Web 数据管理是此框架的组件。A.5 节讨论从特定实例化框架构建信息系统。A.6 节讨论我们已经写过（或正在撰写）CRC Press 中的各图书之间的关系。本附录总结在 A.7 节。

A.2　数据库系统的发展

图 A.1 概述了数据库系统技术的发展。虽然 20 世纪 60 年代早期的工作着重于基于网络和分层数据模型开发产品，但数据库系统的大部分发展，是在 Codd 的原创论文描述关系模型（见本附录参考文献［CODD70］、［DATE90］）之后发生的。关系数据库系统的研究和开发工作在 20 世纪 70 年代初期进行，并且在 70 年代开发了几个原型。显著的成就包括 IBM System R 和加州大学伯克利分校的 INGRES。在 20 世纪 80 年代，许多关系数据库系统的产品上市（其中著名的是 Oracle Corporation，Sybase Inc.，Informix Corporation，INGRES Corporation，IBM，Digital Equipment Corporation 和 Hewlett Packard Company）。在 20 世纪 90 年代，出现了其他厂商的产品（如微软公司）。事实上，到目前为止，更多关系数据库系统的产品已经上市。然而，Codd 表示，许多作为关系系统上市的产品并不是真正的关系系统（参见本附录参考文献［DATE90］中的讨论）。然后，他讨论了系统作为关系数据库系统必须满足的各种标准。早期的工作重点是数据模型、规范化理论、查询处理和优化策略、查询语言、访问策略和索引等问题，后来转向支持多用户环境。特别是开发并发控制和恢复技术。还提供了事务处理支持。

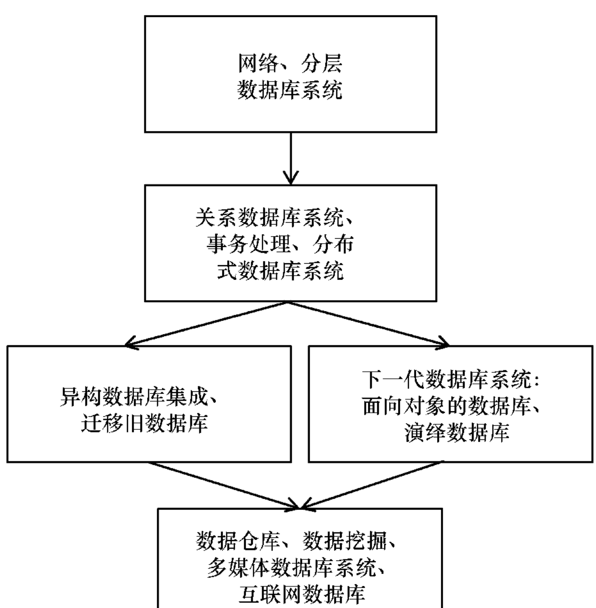


图 A.1　数据库系统技术的发展

关于关系数据库系统以及事务管理的研究，是在20世纪70年代中期分布式数据库系统的研究之后进行的。几个分布式数据库系统的原型开发工作也是在20世纪70年代末开始的。其中包括IBM的System R*、Honeywell Inc.的DDTS（分布式数据库测试台系统）、CCA（美国计算机公司）的SDD-I Multibase和SDC（系统开发公司）的Mermaid。此外，许多这样的系统（例如，DDTS、Multibase、Mermaid）在异构环境中发挥了作用。在20世纪90年代初期，一些数据库系统提供商（如Oracle Corporation、Sybase Inc.、Informix Corporation）为其系统提供了数据分发功能。大多数分布式关系数据库系统的产品均基于客户端—服务器架构。这个想法是让提供商A的客户端与提供商B的服务器数据库系统进行通信。换句话说，客户端—服务器计算模式有助于异构计算环境。关系和非关系商业数据库系统之间的互操作性也是可能的。数据库系统社群也参与标准化工作。在标准化工作中，著名的是ANSI/SPARC 3级架构、数据字典系统的IRDS（信息资源字典系统）标准、关系查询语言SQL（结构化查询语言）和用于远程数据库访问的RDA（远程数据库访问）协议。

数据库技术的另一重大发展，就是面向对象的数据库管理系统的出现。开发这种系统的工作始于20世纪80年代中期，它们现在可以通过商业化方式获得（其中包括Object Design Inc.、Ontos Inc.、Gemstone Systems Inc.、Versant Object Technology的产品）。有人认为，新一代应用如多媒体、办公信息系统、CAD/CAM、过程控制和软件工程有着不同的要求。这样的应用使用复杂的数据结构。编程语言和数据模型之间更紧密的集成也是需要的。面向对象的数据库系统满足了这些新一代应用程序的大部分要求（见本附录参考文献［CATT91］）。

根据1990年NSF研讨会发布的Lagunita报告（见本附录参考文献［SILB90］和［KIM90］），关系数据库系统、事务处理和分布式（关系）数据库系统被称为成熟技术。此外，提供商正在推广面向对象的数据库系统，并展示不同数据库系统之间的互操作性。该报告继续说明，随着应用程序越来越复杂，需要更复杂的数据库系统。此外，由于许多组织现在使用数据库系统，在许多不同类型的情况下，就需要集成数据库系统。虽然一些研究已经开始解决这些问题，商业产品已经可以使用，但仍有几个需要解决的问题。因此，数据库系统研究人员在20世纪90年代初面临的挑战分为两个方面，一个是下一代数据库系统，另一个是异构数据库系统。

下一代数据库系统包括面向对象的数据库系统、功能数据库系统、增强数据库系统性能的特殊并行架构、高性能数据库系统、实时数据库系统、科学数据库系统、时态数据库系统、处理不完整和不确定信息的数据库系统，以及智能数据库系统（有时也称为逻辑或演绎数据库系统）。理想情况下，数据库系统应提供对高性能事务处理、模型复杂应用程序、代表新类型数据的支持，并进行智能演绎。虽然在20世纪80年代末和90年代初期取得了重大进展，但在开发这种数据库系统之前，还有很多工作要做。

异构数据库系统在过去十年中一直受到相当大的关注（见本附录参考文献［MARC90］）。其中的主要问题包括处理不同的数据模型、不同的查询处理策略、不同的事务处理算法和不同的查询语言。应该对整个系统提供统一的视图，还是由单个系统

的用户来维持对整个系统的视图？这些问题尚未得到令人满意的回答。我们只是设想，一个完整的异构数据库管理系统解决方案，也许需要几代人的势力。后续研究应该是寻找这样针对性的解决方案，还应该开展工作来处理有限形式的异构性，以满足客户的需求。最近一直受到关注的另一种类型的数据库系统，是联盟数据库系统。需要注意的是，有些用户可以互换使用异构数据库系统和联盟数据库系统。异构数据库系统是联盟的一部分，联盟也包括同构数据库系统。

网络用户的爆炸式增长以及接口技术的发展，为数据管理研究人员带来了更多的挑战。第二次研讨会由 NSF 在 1995 年发起，几项新兴技术在我们进入 21 世纪时已被认为是重要的（见本附录参考文献［WIDO96］）。这包括数字图书馆、管理非常大的数据库、数据管理问题、多媒体数据库、数据仓库、数据挖掘、协作计算环境的数据管理以及安全和隐私。20 世纪 90 年代的另一个重大发展是对象关系系统的发展。这样的系统结合了面向对象数据库系统和关系数据库系统的优点。此外，许多公司现在的重点是将其数据管理产品与网络技术相结合。最后，对许多组织来说，越来越需要将一些遗留数据库和应用程序迁移到较新的架构和系统中，如客户端—服务器体系结构和关系数据库系统。我们相信数据管理系统的研发并没有结束。随着新技术的发展，数据管理的研究与开发也将有新的机遇。

所有数据管理技术的综合视图如图 A.2 所示。图中显示，传统技术包括数据库设计、事务处理和基准测试。数据库系统基于关系和面向对象的数据模型。数据库系统可能依赖于他们所提供的功能。例如，安全性和实时性。这些数据库系统也可能是关系的或面向对象的，还有基于多个站点或处理器的数据库系统。例如，分布式和异构数据库系统、并行系统和正在迁移的系统。最后，还有数据仓储和挖掘、协作和网络等新兴技术。关于数据管理系统的任何综合文献都应该涉及所有这些技术。我们选择了一些相关技术，并将其纳入框架。该框架在 A.5 节中有描述。

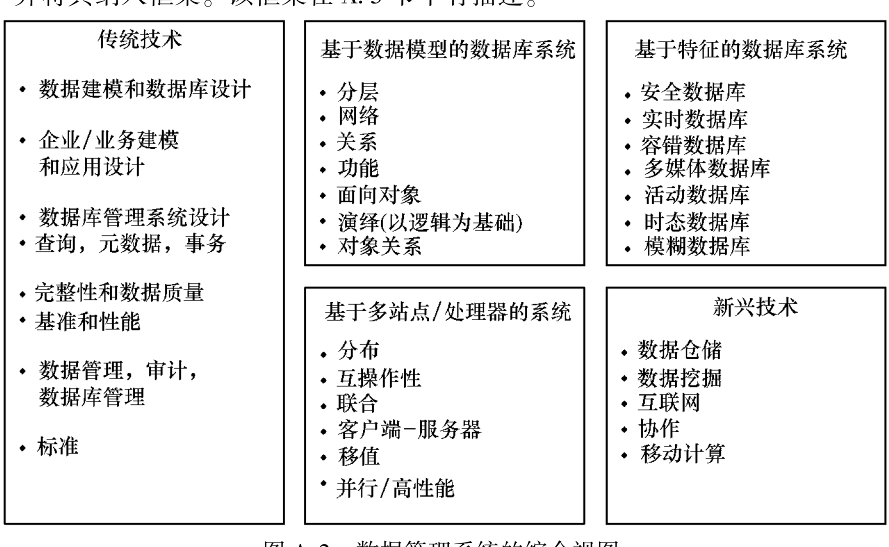


图 A.2　数据管理系统的综合视图

A.3 地位、愿景和问题

数据管理系统取得了重大进展。然而，许多技术仍然是独立的技术，如图 A.3 所示。例如，多媒体系统尚未成功地与仓储和挖掘技术相结合。我们最终的目标是集成多种技术，以便在适当的时间产生准确的数据和信息，并及时分发给用户。我们的数据和信息管理愿景如图 A.4 所示。

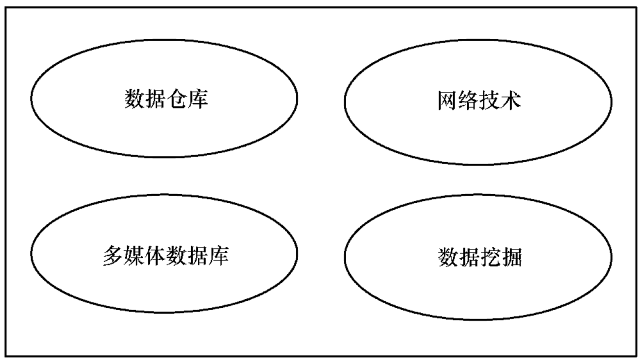


图 A.3 单机系统

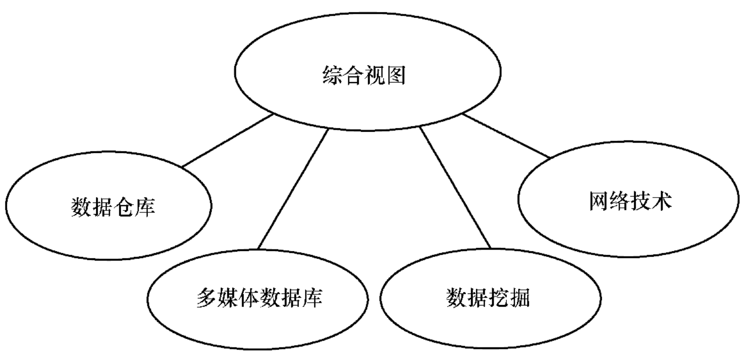


图 A.4 愿景

本附录参考文献［THUR97］中讨论的工作解决了实现这一愿景面临的许多挑战。特别地，讨论了异构数据库的集成，以及分布式对象技术在互操作性方面的应用。虽然在互操作性的系统方面取得了很大进展，但语义问题仍然是一个挑战。不同的数据库有不同的表示。此外，可以在不同的站点对相同的数据实体进行不同的解释。解决这些语义差异，并从异构和可能的多媒体数据源中提取有用的信息将是主要的问题。本书尝试通过使用数据挖掘来解决一些相关问题。

A.4 数据管理系统框架

为了成功地开发可互操作的数据管理系统，异构的数据库系统集成是一个主要的组件。然而，还必须有其他技术顺利地相互融合，以开发有效访问数据和共享技术，以及

从数据中提取信息。为了促进数据管理系统的发展，以满足医疗、金融、制造、军事等领域各种应用的需求，我们提出了一个框架，可以作为数据管理系统的参考模型。来自该框架的各种组件必须被集成，以开发数据管理系统，并能支持各种应用。

图 A. 5 说明了我们的框架，可以被视为数据管理系统中的一个模型。这个框架由三层组成。可以想到组件技术，我们也将其称为组件，属于特定层，并或多或少地构建在下层所提供的技术之上。第一层是数据库技术和分布层。该层由数据库系统和分布式数据库系统技术组成。

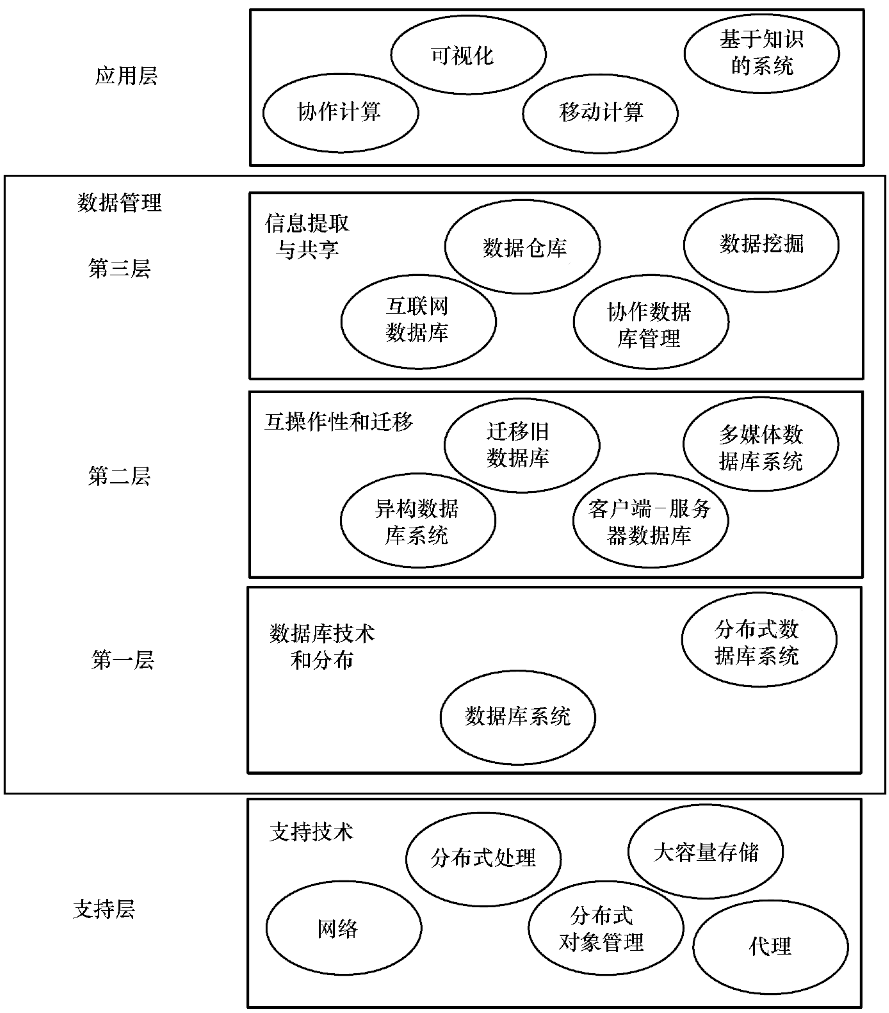


图 A. 5　数据管理系统框架

第二层是互操作性和迁移层。该层由如异构数据库集成、客户端–服务器数据库和处理异构数据类型的多媒体数据库系统，以及迁移旧数据库等技术组成。第三层是信息提取与共享层。该层本质上由数据管理系统所支持的一些较新服务的技术组成。这些技

术包括数据仓库、数据挖掘（见本附录参考文献［THUR98］）、Web 数据库和协作应用程序的数据库支持。数据管理系统可以利用如网络、分布式处理和大容量存储之类的较低级技术。我们将这些技术分组到支持层中。该支持层不属于数据管理系统框架，该层包括一些更高级别的技术，如分布式对象管理和代理。另外，如图 A. 5 所示，还有应用层。属于应用层的，如协同计算系统和基于知识的系统等类似可以利用数据管理的系统。需要注意的是，应用层也不在数据管理系统框架之内。

构成数据管理系统框架的技术，可以被认为是数据管理中的一些核心技术。然而，许多使用数据管理技术的应用程序都需要安全性、完整性、实时处理、容错和高性能计算等功能。利用数据管理技术的应用可能是医疗、金融或军事等。我们在图 A. 6 中进行了说明，给出了将数据管理技术与特征和应用相关联的三维视图。例如，可以开发用于医疗应用的安全分布式数据库管理系统，或用于金融领域的容错多媒体数据库管理系统。

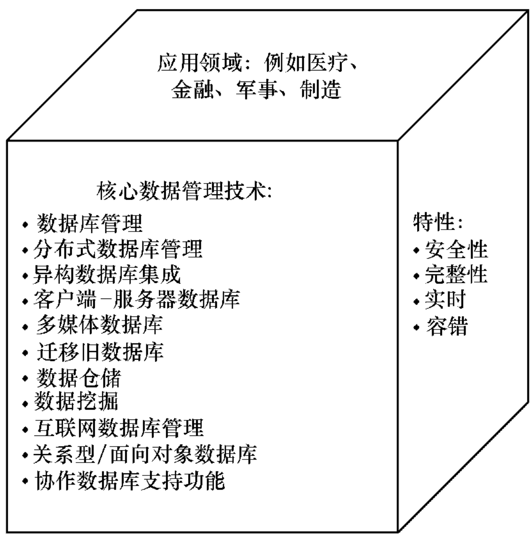


图 A. 6　数据管理的三维视图

集成属于各层的组件，对于开发高效的数据管理系统很重要。此外，数据管理技术必须与应用技术相结合，以开发成功的信息系统。但是，目前这些组件之间的集成是有限的。我们以前的“数据管理系统演进与互操作”（见本附录参考文献［THUR97］），主要集中在框架中所显示的每个组件的概念、发展和趋势上。此外，我们目前的关于 Web 数据管理书籍（我们也称为 Web 数据管理）专注于图 A. 5 框架中第 3 层的 Web 数据库组件。需要注意的是，安全性可跨越所有层。支持层需要安全性，如代理和分布式系统。架构中的所有层都需要安全性，包括数据库安全性、分布式数据库安全性、仓储安全性、Web 数据库安全性和协作数据管理安全性。这是本书的主题。也就是说，我们涵盖了数据和应用程序安全性的各个方面，包括数据库安全和信息管理安全。

A. 5　构建基于框架的信息系统

图 A. 5 所示为数据管理系统的框架。如图所示，数据管理技术包括数据库系统、

分布式数据库系统、异构数据库系统、迁移旧数据库、多媒体数据库系统、数据仓库、数据挖掘、Web 数据库和协作数据库支持。此外，数据管理系统可利用分布式处理和代理等支持技术。类似地，如协同计算、可视化、专家系统和移动计算等都要利用数据管理系统。

我们中的许多人都曾经多次听说过信息系统这个术语。这些系统有时可与数据管理系统互换使用。在我们的术语中，信息系统比数据管理系统更广泛，而且它包括数据管理系统。事实上，信息系统的框架不仅包括数据管理系统层，还包括支持层以及应用层。也就是说，信息系统包括各种计算系统。可以视为可用于各种应用的成品。也就是说，硬件处于最低端，应用程序处于最高端。

我们可以结合图 A.5 的技术将信息系统放在一起。例如，在应用程序层面，可能需要协作和可视化技术，以便分析人员可以协作地执行一些任务。在数据管理层面，可能需要多媒体和分布式数据库技术。在支持层面，可能需要海量存储以及一些分布式处理能力。这一特殊框架如图 A.7 所示。另一个例子是互操作性的特殊框架。可能需要一些可视化技术来显示来自异构数据库的集成信息。在数据管理层面，我们有异构的数据库系统技术。在支持层面，可以使用分布式对象管理技术来封装异构数据库。这一特殊框架如图 A.8 所示。

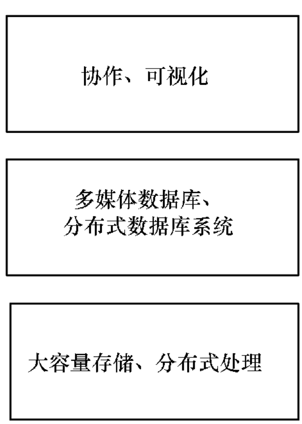


图 A.7　协作多媒体数据管理框架



图 A.8　异构数据库互操作框架

最后，让我们用一个具体的例子，来说明上面我们描述过的概念。假设一群医生/外科医生想要一个系统，在系统中他们可以协作和诊断各种患者。这可能是医疗视频电话会议应用程序。也就是说，在最高级别，该应用是医疗应用，更具体地说，是医疗视频电话会议应用。在应用层面，需要多种技术，包括协作和电话会议。这些应用技术将利用数据管理技术，如分布式数据库系统和多媒体数据库系统。也就是说，可能需要支持音频和视频等多媒体数据。数据管理技术又利用了较低级别的技术，如分布式处理和网络化。我们在图 A.9 中进行了说明。

虽然应用技术利用数据管理技术，而数据管理技术利用支持技术，但信息系统的最

终用户还是应用程序本身。今天许多应用程序利用信息系统。这些应用来自医疗、金融、制造、电信和国防等多个领域，具体应用包括信号处理、电子商务、病人监护和情况评估。图 A. 10 说明了应用程序与信息系统之间的关系。

总而言之，信息系统包括数据管理系统，以及如协作计算系统的应用层系统和如分布式对象管理系统等支持层的系统。

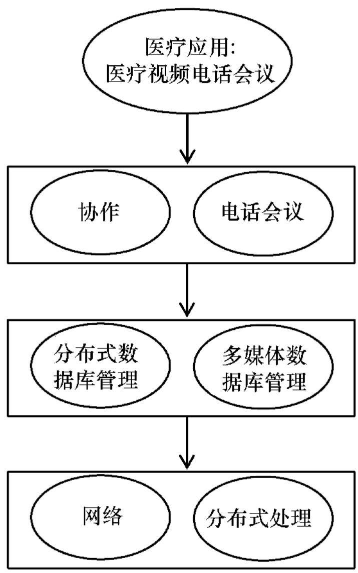


图 A. 9　具体示例

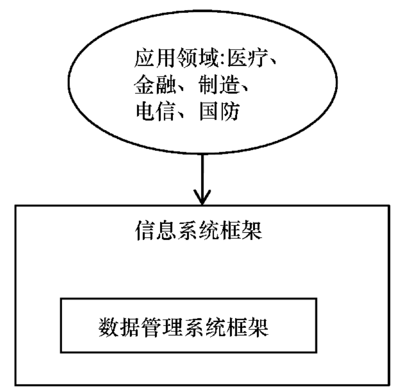


图 A. 10　应用—框架关系

A. 6　书籍之间的关系

我们已经出版了两个系列的丛书。第一系列主要针对技术管理人员，而第二个系列则针对研究人员和开发人员。本书是第一个系列中的第十个。我们以前的九本书是《数据管理系统演化和互操作》（见本附录参考文献［THUR97］）、《数据挖掘：技术、工具和趋势》（见本附录参考文献［THUR98］）、《网络数据管理和电子商务》（见本附录参考文献［THUR00］）、《为电子企业管理和挖掘多媒体数据库》（见本附录参考文献［THUR01］）、《XML、数据库和语义 Web》（见本附录参考文献［THUR02］）、《Web 数据挖掘及其在商业智能和反恐中的应用》（见本附录参考文献［THUR03］）、《数据库和应用程序安全：集成数据管理和信息安全》（见本附录参考文献［THUR05］）、《建立可信赖的语义 Web》（见本附录参考文献［THUR07］）和《面向语义服务的安全系统》（见本附录参考文献［THUR10］）。本书（见本附录参考文献［THUR13］）是从我们以前的《面向语义服务的安全系统》一书中演变而来。所有这些书籍都是从我们在本附录中说明的框架中演变而来的，并且还涉及框架的不同部分。这些书籍之间的联系如图 A. 11 所示。

我们已经出版了两本关于第二辑的书。第一个是《数据挖掘工具的设计和实现》（见本附录参考文献［AWAD09］），其次是《恶意软件检测的数据挖掘工具》（见本附录参考文献［MASU11］），其内容关联性如图 A. 12 所示。

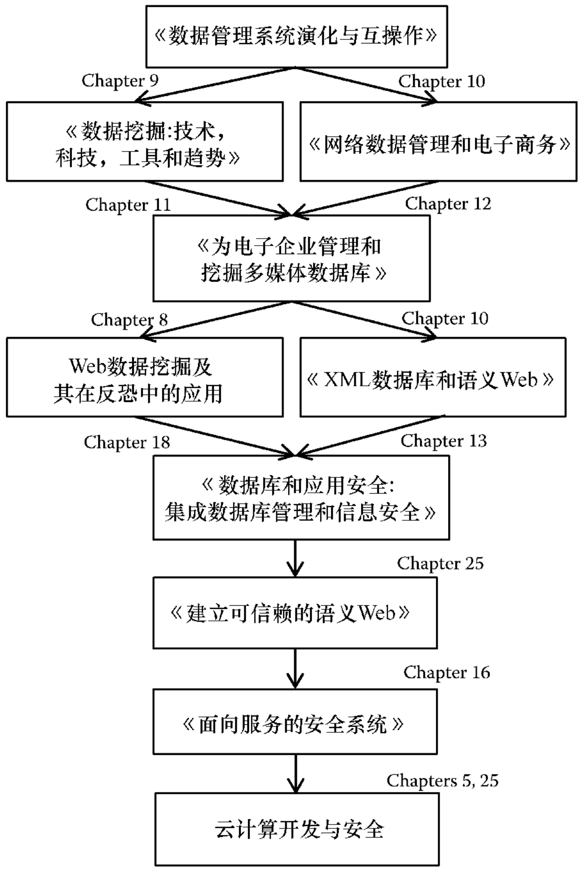


图 A. 11　书籍之间的关系 – 系列 I

A. 7　总结和展望

在附录 A 中，我们提供了数据管理的概述。首先，我们讨论了数据管理的发展，然后提出了数据管理的愿景。然后，我们说明了一个数据管理框架。该框架由 3 层组成：数据库系统层、互操作层和信息提取层。Web 数据管理属于第 3 层。最后，我们展示了如何由框架技术来构建信息系统。

我们认为数据管理对许多信息技术至关重要，包括数据挖掘、多媒体信息处理、互操作性以及协作和知识管理。在本附录中强调数据管理。安全性对于所有数据管理技术都至关重要，我们依靠这些技术来实现云计算。我们将在附录 B 中提供有关数据挖掘的背景信息。附录 C 将提供数据管理中的访问控制背景知识。我们将在附录 D 中讨论我们保障信息共享的方法。

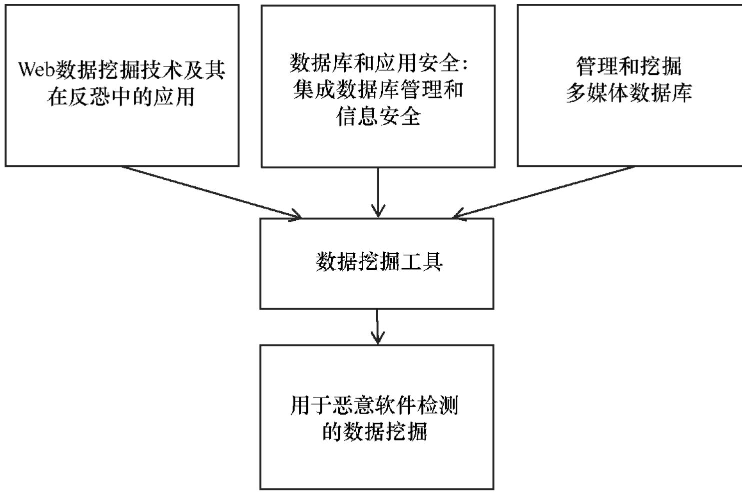


图 A. 12　书籍之间的关系 – 系列 Ⅱ

参 考 文 献

[AWAD09] Awad, M., L. Khan, B. Thuraisingham, and L. Wang, *Design and Implementation of Data Mining Tools*, CRC Press, Boca Raton, FL, 2009.

[CATT91] Cattell, R., *Object Data Management Systems*, Addison- Wesley, MA, 1991.

[CODD70] Codd, E. F., A relational model of data for large shared data banks. *Communications of the ACM*, 13(6): 377–387, June 1970.

[DATE90] Date, C. J., *An Introduction to Database Management Systems*. Addison-Wesley, MA, 1990 (8th edition, 2003, MA).

[KIM90] Kim, W. (Ed.), *Directions for Future Database Research & Development*. ACM SIGMOD Record, December 1990, 19(4).

[MARC90] March, S. T. (Ed.), *Special Issue on Heterogeneous Database Systems*, ACM Computing Surveys, September 1990, 22(3).

[MASU11] Masud, M., B. Thuraisingham, and L. Khan, *Data Mining Tools for Malware Detection*, CRC Press, Boca Raton, FL, 2011.

[SILB90] Silberschatz, A., Stonebraker, M., and Ullman, J.D. (Eds) (1990), *Database Systems: Achievements and Opportunities, The Lagunita Report of the NSF Invitational Workshop on the Future of Database Systems Research,* February 22–23, Palo Alto, CA (TR-90-22), Department of Computer Sciences, University of Texas at Austin, Austin, TX (also in *Communications of the ACM*, 34(10), October 1991, 110–120).

[THUR97] Thuraisingham, B., *Data Management Systems: Evolution and Interoperation*, CRC Press, Boca Raton, FL, 1997.

[THUR98] Thuraisingham, B., *Data Mining: Technologies, Techniques, Tools and Trends*, CRC Press, Boca Raton, FL, 1998.

[THUR00] Thuraisingham, B., *Web Data Management and Electronic Commerce*, CRC Press, Boca Raton, FL, 2000.

[THUR01] Thuraisingham, B., *Managing and Mining Multimedia Databases for the Electronic Enterprise*, CRC Press, Boca Raton, FL, 2001.

[THUR02] Thuraisingham, B., *XML, Databases and the Semantic Web*, CRC Press, Boca Raton, FL, 2002.
[THUR03] Thuraisingham, B., *Web Data Mining Applications in Business Intelligence and Counter-terrorism*, CRC Press, Boca Raton, FL, 2003.
[THUR05] Thuraisingham, B., *Database and Applications Security: Integrating Data Management and Information Security*, CRC Press, Boca Raton, FL, 2005.
[THUR07] Thuraisingham, B., *Building Trustworthy Semantic Webs*, CRC Press, Boca Raton, FL, 2007.
[THUR10] Thuraisingham, B., *Secure Semantic Service-Oriented Systems*, CRC Press, Boca Raton, FL, 2010.
[THUR13] Thuraisingham, B., *Building and Securing the Cloud*, CRC Press, Boca Raton, FL, 2013.
[WIDO96] Widom, J., (Ed.), Database research: Achievements and opportunities into the 21st century. *Proceedings of the Database Systems Workshop, Report Published by the National Science Foundation, 1995*, 25(1), (also in ACM SIGMOD Record, March 1996).

附录 B　数据挖掘技术

B. 1　概述

数据挖掘的输出（也称为任务）包括分类、聚类、形成关联以及检测异常。我们的工具主要关注分类的结果，我们已经开发了分类工具。分类问题也被称为监督学习，其中通过模型学习一组标记的示例，然后将具有未知标签的新示例呈现给模型，从而用于预测。

有许多预测模型已经投入实际使用中，如马尔科夫模型、决策树、人工神经网络（Artificial Neural Networks，ANN）、支持向量机（Support Vector Machines SVM）、关联规则挖掘（Association Rule Mining，ARM）等。这些模型都有自己的优点和缺点。然而，在所有这些技术中存在着无法适应所有应用的共同弱点。没有相应理想的或完美的分类的原因是，每种技术最初均是为了在某些假设情况下解决具体问题而设计的。

在本附录中，我们将讨论在工具中使用的数据挖掘技术。具体来说，我们提出了马尔可夫模型、SVM、ANN、ARM，还有多分类的问题以及图像分类，这是图像挖掘的一个方面。我们开发了混合模型，以提高各种应用的数据挖掘预测精度，即入侵检测、网页预测和图像分类。本书讨论了云中托管的一些其他应用程序。

本附录的结构如下。在 B. 2 节中，我们对各种数据挖掘任务和技术进行概述。与本书内容相关的技术在 B. 2 至 B. 6 节中讨论。特别是，对神经网络、SVM、马尔可夫模型和 ARM，以及一些其他的分类技术进行了说明。本附录总结在 B. 7 节。

B. 2　数据挖掘任务和技术概述

在讨论数据挖掘技术之前，我们提供了一些数据挖掘任务（也称为数据挖掘输出）的概述，然后我们再来讨论这些技术。一般来说，数据挖掘任务可以分为两类：预测和

描述。预测任务基本上是预测项目是否属于一类，描述任务一般是从示例中提取其模式。最突出的预测任务之一是分类。在某些情况下，其他任务，像异常检测就可以简化为预测任务，预测某一特定情况其是否异常。描述任务一般包括建立关联和形成集群。因此，分类、异常检测、关联和形成集群均可被认为是数据挖掘任务。

数据挖掘技术可以是预测性的、描述性的或两者兼有。例如，神经网络可以执行分类以及聚类。分类技术包括决策树、SVM 以及基于记忆的推理。通常使用 ARM 技术进行关联。链接分析也可以建立链接之间的关联，并预测新的链接。聚类技术包括 K 均值聚类。数据挖掘任务的概述（即数据挖掘的结果）如图 B.1 所示。本附录中将要讲述的技术（如神经网络、SVM）如图 B.2 所示。

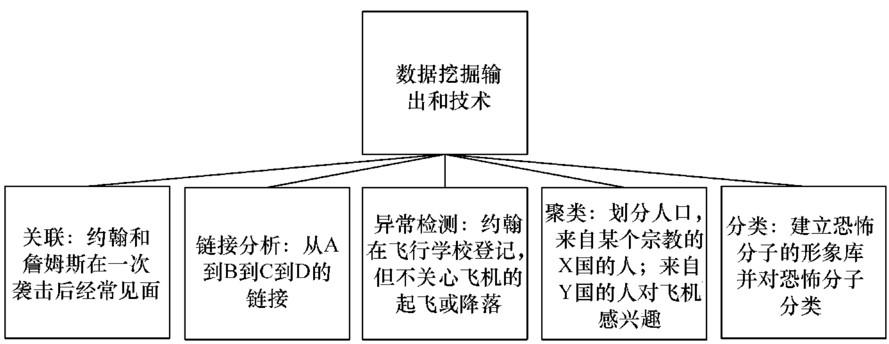


图 B.1　数据挖掘任务

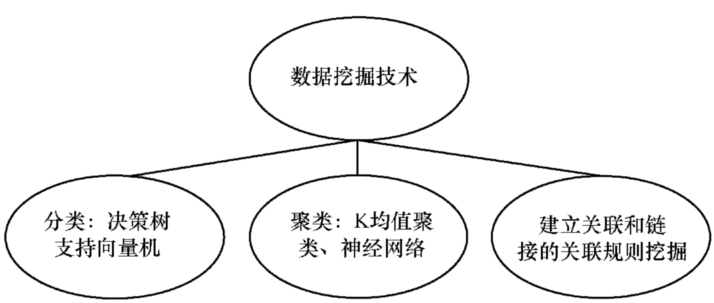


图 B.2　数据挖掘技术中使用的工具

B.3　人工神经网络

ANN 是一种非常有名的、强大的、健壮的分类技术，用于近似实值、离散值和向量值函数。它已经被用于许多领域中。例如，解释视觉场景、语音识别和学习机器人控制策略。ANN 模拟人脑中的生物神经系统。这种神经系统由大量高度相互联系的处理单

位（神经元）组成，它们共同协作，产生我们的感受和反应。ANN 像人一样学习。人脑中的学习过程涉及对神经元之间突触连接的调整。类似地，ANN 的学习过程涉及对节点权重的调整。图 B.3 显示了一个简单的神经元单元，称为感知器。感知器输入 $\boldsymbol{x}$ 是向量或实值输入。$\boldsymbol{w}$ 是权重向量，其值在训练后确定。感知器计算输入向量 $\boldsymbol{x}$ 的线性组合［见式（B.1）］。如下

$$o(x_1,\cdots,x_n)=\begin{cases}1 & 若\ w_0+w_1x_1+\cdots+w_nx_n>0\\ -1 & 其他\end{cases}\tag{B.1}$$

注意，w_i对应感知器输出的输入向量分量 x_i的贡献。此外，为了感知器输出 1，输入（ $\sum_{i=1}^{n} w_i x_i$ ）的加权组合必须大于阈值 w_0。

学习感知器涉及选择权重的值 $w_0+w_1x_1+\cdots+w_nx_n$ 。最初，感知器被赋予随机权重值。

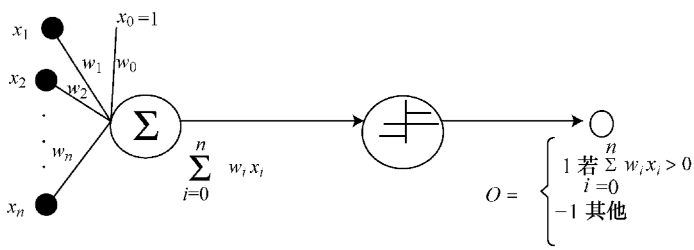


图 B.3　感知器

然后感知器被应用到每一个训练实例中，每当一个例子被错误分类时，就会更新感知器的权重。这个过程重复很多次，直到所有的训练样本被正确分类。根据以下规则更新权重［见式（B.2）］：

$$\begin{cases}w_i=w_i+\delta w_i\\ \delta w_i=\eta(t-o)x_i\end{cases}\tag{B.2}$$

式中，η 是学习常数；o 是感知器计算的输出；t 是当前训练样本的目标输出。

单个感知器的计算能力限于线性决策。然而，感知器可以作为强大的多层网络中的组成部分。在这种情况下，需要更复杂的更新规则来训练网络权重。在这项工作中，我们采用两层 ANN，每层均由 3 个构建块组成（见图 B.4）。我们使用反向传播算法来学习权重。反向传播算法尝试使用最小化二次方误差函数。

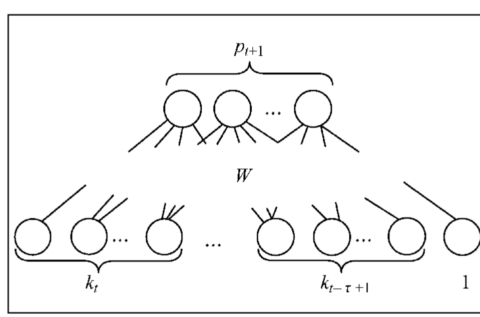


图 B.4　人工神经网络

WWW 预测中的一个典型的训练示例是 $<[k_{t-\tau+1},\cdots,k_{t-1},k_t]^{\mathrm{T}},d>$，其中 $[k_{t-\tau+1},\cdots,k_{t-1},k_t]^{\mathrm{T}}$ 是 ANN 的输入，d 是目标网页。需要注意的是，图 B.5 中 ANN 的输入单元是用户最近访问过的 τ 先前页，其中 k 是网页 ID。网络的输出是一个布尔值，而不是概率。稍后我们将看到如何通过在 ANN 输出后拟合 Sigmoid 函数来近似输出概率。近似的概率输出变为 $o' = f(o(I)) = p_{t+1}$，其中 I 是输入会话，$p_{t+1} = p(d \mid k_{t-\tau+1},\cdots,k_t)$ 我们选择 Sigmoid 函数［见式（B.3）］作为传递函数，以便 ANN 可以处理非线性可分离数据（见本附录参考文献［MITC97］）。应注意到，在我们的 ANN 设计（见图 B.5）中，我们在每个构建块中均使用 Sigmoid 传递函数。在式（B.3）中，I 是网络的输入，o 是网络的输出，w 是权重矩阵，σ 是 Sigmoid 函数。

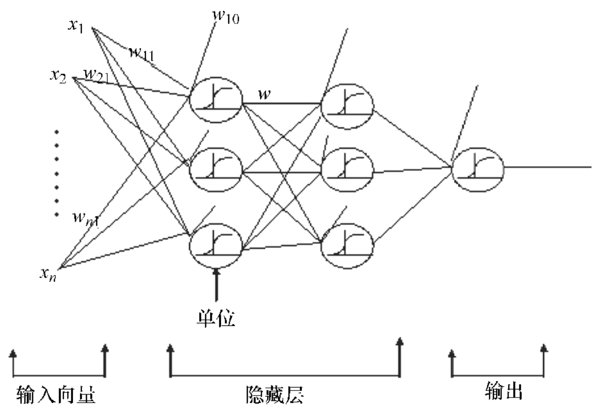


图 B.5　人工神经网络的设计在我们的实验中的应用

$$o = \sigma(w \cdot I)$$
$$\sigma(y) = \frac{1}{1+\mathrm{e}^{-y}} \tag{B.3}$$

$$E(W) = \frac{1}{2}\sum_{k\in D}\sum_{i\in output}(t_{ik} - o_{ik})^2 \tag{B.4}$$

$$\begin{cases} w_{ji} = w_{ji} + \delta w_{ji} \\ \delta w_{ji} = -\eta \dfrac{\partial E_{\mathrm{d}}}{\partial w_{ji}} \end{cases} \tag{B.5}$$

$$\delta w_{ji}(n) = -\eta \frac{\partial E_{\mathrm{d}}}{\partial w_{ji}} + \alpha \delta w_{ji}(n-1) \tag{B.6}$$

我们实施用于训练权重的反向传播算法。反向传播算法采用梯度下降法，以试图最小化网络的输出值和这些输出的目标值之间的二次方误差。所有网络输出单元的误差之和在式（B.4）中定义。在式（B.4）中，输出是网络中的输出单元集合，D 是训练集，t_{ik}和 o_{ik}是与第 i 个输出单元和训练样本 k 相关联的目标和输出值。对于网络中的特定权重 w_{ji}，以及对于每个训练示例，如式（B.5）所示，对其进行更新，其中 η 是学习速率，w_{ji}是与网络单元 j 的第 i 个输入相关联的权重（详见本附录参考文献［MITC97］）。从式（B.5）可以看出，使用梯度下降法计算搜索方向 δw，该梯度下降法保证向局部最小值收敛。为了减缓收敛，我们向权重更新规则添加动量，使得权重更新方向 $\delta w_{ji}(n)$ 部分地取决于上一次迭代 $\delta w_{ji}(n-1)$ 中的更新方向。新的权重中的更新方向如式（B.6）所示，其中 n 是当前迭代，α 是动量常数。需要注意的是，在式（B.6）中，步

长略大于式（B.5）。这有助于在梯度不变的区域中搜索能平滑收敛（见本附录参考文献［MITC97］）。

在我们的实验中，我们基于数据集中类的分布来动态地设置步长 η。具体来说，当更新属于低分布类的训练样本时，我们将步长设置为大值，反之亦然。这是因为当数据集中类的分布变化很大时（例如，数据集可能具有 5% 的正例和 95% 的负例），则网络权重就从较大分布的类中收敛到示例，这将导致缓慢收敛。此外，我们通过应用动量常数［见式（B.6）］来略微调整学习率，以加快网络的收敛（见本附录参考文献［MITC97］）。

B.4　支持向量机

SVM 是一个学习系统，它在一个高维度的特征空间中使用线性函数的假设空间，用最优化理论的学习算法进行训练。Vapnik 等人引进的这种学习策略（见本附录参考文献［CRIS00］）是一种非常好的方法，已被广泛应用于各种应用。SVM 中的基本概念是超平面分类器或线性可分性。为了实现线性可分性，SVM 应用两个基本思想：边距最大化和内核。即，将输入空间映射到更高维度的空间、特征空间。

对于二进制分类，SVM 问题可以形式化为式（B.7）。假设我们有 N 个训练数据点 $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，其中 $x_i R^d$ 和 $y_i \in \{+1, -1\}$。我们想找到线性分离超平面分类器如式（B.8）所示。此外，我们希望这个超平面相对于两个类来说，具有最大的分离边距（见图 B.6）。函数边距或短边距在几何上被定义为，从决策边界到输入空间最近点的欧几里得距离。图 B.7 说明了为什么边距最大化能给出最佳分离解决方案的直观解释。在图 B.7 的 A 部分中，我们可以找到特定数据集中无限数量的分隔符。没有明确的理由证实一个分离器超过另一个分离器。在 B 部分中，我们看到，最大化边距只能提供一个厚度分离器。这样的解决方案证明可以达到最佳的泛化精度，即对于未知的预测（见本附录参考文献［VAPN95］、［VAPN98］、［VAPN99］）。

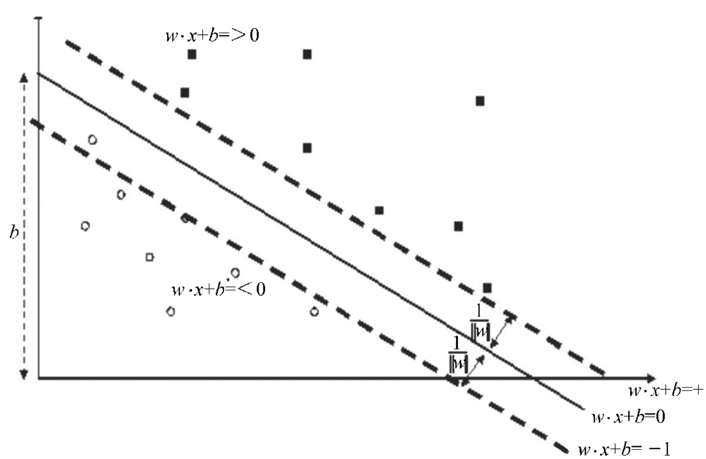


图 B.6　SVM 中的线性分离

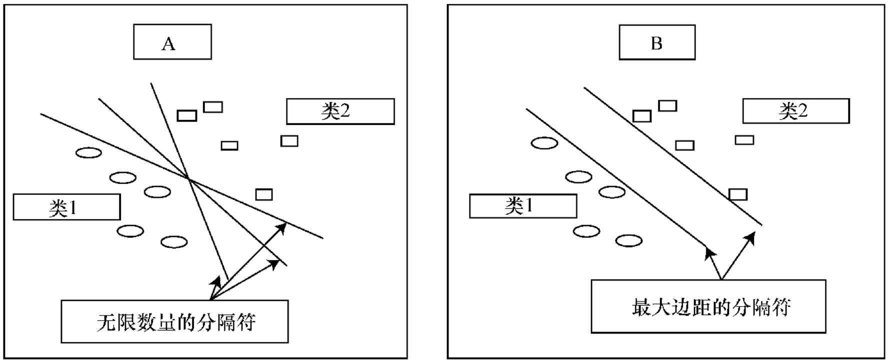


图 B.7　支持最大边距的 SVM 分隔符

$$\begin{cases}\text{minimize}_{(w,b)}\dfrac{1}{2}w^{\mathrm{T}}w\\ \text{subject to } y_i(w\cdot x_i-b)\geqslant 1\end{cases}\tag{B.7}$$

$$f(x)=\text{sign}(w\cdot x-b)\tag{B.8}$$

$$\text{maximize}L(w,b,\alpha)=\frac{1}{2}w^{\mathrm{T}}w-\sum_{i=1}^{N}\alpha_i y_i(w\cdot x_i-b)+\sum_{i=1}^{N}\alpha_i\tag{B.9}$$

$$f(x)=\text{sign}(wx-b)=\text{sign}\left[\sum_{i=1}^{N}\alpha_i y_i(x\cdot x_i-b)\right]\tag{B.10}$$

应注意到，式（B.8）除了有 x 的预测标签之外，还有计算点 x 的功能边距的符号，即 x 的函数边距等于 $wx-b$。

SVM 优化问题是一个凸二次规划问题（在 w，b）中的凸集方程式（B.7）。相对于 α，我们可以像式（B.9）那样解决沃尔夫二进制，受到 $L(w, b, \alpha)$ 相对于原始变量 w 和 b 的梯度消失的约束，$\alpha_i\geqslant 0$。原始变量从 $L(w, b, \alpha)$ 中消除（详见本附录参考文献［CRIS00］）。当我们求解 α_i 时，我们可以得到 $w=\sum_{t=1}^{N}=\alpha_i y_i x_i$，我们可以使用式（B.10）对一个新对象 x 进行分类。需要注意到，训练矢量仅以点积的形式出现，并且每个训练点存在拉格朗日乘数 α_i，这反映了数据点的重要性。当找到最大边界超平面时，只有最靠近超平面的点将具有 $\alpha_i=0$ 特征，这些点称为支持向量。所有其他点将具有 $\alpha_i=0$ 特征（见图 B.8a）。这意味着只有最靠近超平面的点才能给出假设/分类器的表示。这些最重要的数据点作为支持向量。它们的值也可用于给出关于假设/分类器的可靠性的独立边界（见本附录参考文献［BART99］）。

图 B.8a 显示了两个类及其边界，即边距。支持向量由实体对象表示，而空对象是非支持向量。需要注意的是，边距仅受支持向量的影响。也就是说，如果我们删除或添加空对象，则边距就不会改变。同时，固体对象的任何更改（添加或删除对象）都可能会更改边距。图 B.8b 显示了在边缘区域添加对象的效果。我们可以看到，添加或删除远离边的对象，例如数据点 1 或 −2，不会改动边距。但是，添加和/或删除边附近的对象，例如数据点 2 和/或 −1，就会创建新的边距。

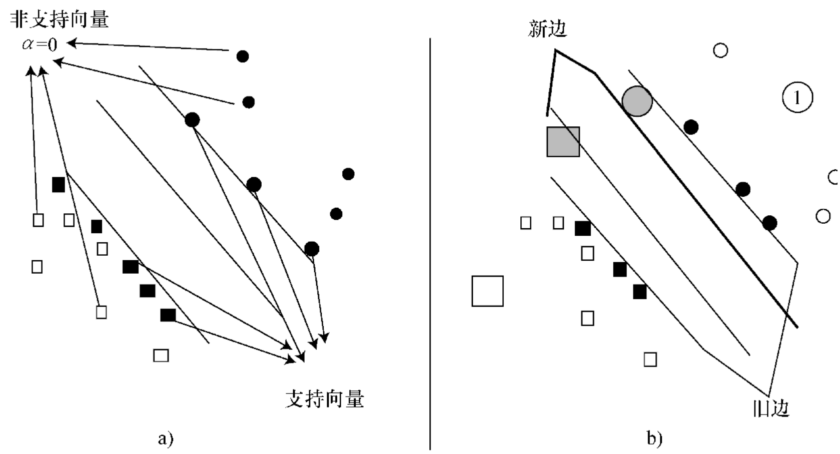


图 B.8　a）支持向量和非支持向量的值　b）增加新数据点的效果

B.5　马尔可夫模型

最新最先进适于网上冲浪的一些预测模型是利用马尔可夫模型来开发的（见本附录参考文献［YANG01］、［PIRO96］）。对于这些预测模型，上网者访问网页中的序列通常被认为是马尔可夫链，然后将其作为输入。马尔可夫模型的基本概念是，它根据以前的动作或动作的结果预测下一个动作。动作可能意味着不同的应用程序中具有不同的东西。为了更好地说明，我们将讲述面向 WWW 预测应用的特定活动。在 WWW 预测中，下一个动作对应于要遍历的下一个页面的预测。以前的动作对应于要考虑的以前的网页。在以前考虑的行动数量的基础上，马尔可夫模型可以有不同的顺序。

$$pr(P_k) = pr(S_k) \tag{B.11}$$

$$pr(P_2 \mid P_1) = pr(S_2 = P_2 \mid S_1 = P_1) \tag{B.12}$$

$$pr(P_N \mid P_{N-1},\cdots,P_{N-k}) = pr(S_N = P_N \mid S_{N-1} = P_{N-1},\cdots,P_{N-k}) \tag{B.13}$$

零阶马尔可夫模型是状态（或网页）的无条件概率［见式（B.11）］。在式（B.11）中，P_k 是网页，S_k 是相应的状态。可以通过采用页面到页面的转移概率或 $\{P_1, P_2\}$，$\{P_2, P_3\}$，⋯，$\{P_{k-1}, P_k\}$ 的 n – gram 概率来计算一阶马尔可夫模型，如式（B.12）。

接下来，我们给出了说明马尔可夫模型不同阶数的例子，以及怎样预测。

范例：

想象一下，网站有 6 个网页：P_1、P_2、P_3、P_4、P_5 和 P_6。假设我们有用户会话，如表 B.1 所示。表 B.1 描述了该网站中许多用户的网页导航。图 B.9 所示为一阶马尔可夫模型，其中根据用户执行的最后一个动作（即最后一页），预测下一个动作。状态 S 和 F 分别对应于初始状态和最终状态。每个转换的概率是通过遍历状态序列次数与锚固状态访问次数的比率来估计的。在图 B.8 中的每个拱形旁边，第一个数字是该转换的

频率，第二个数字是转移概率。例如，由于用户从第 2 页转到第 3 页的访问次数为 3，对第 2 页的访问次数是 15，转换（P_2-P_3）的转移概率为 0.2（即，0.2 = 3/15）。

表 B.1　用户会话及其频率的集合

会话	频率
P_1，P_2，P_4	5
P_1，P_2，P_6	1
P_5，P_2，P_6	6
P_5，P_2，P_3	3

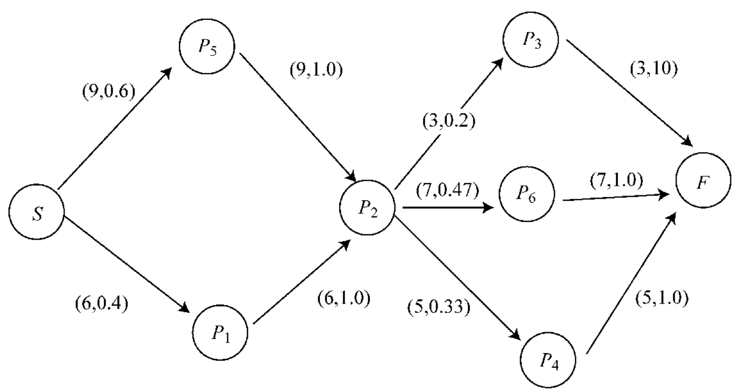


图 B.9　一阶马尔可夫模型

需要注意的是，转移概率用于解决预测问题。例如，假设用户已经访问了 P_2，那么接下来访问的最可能的页面是 P_6。那是因为从 P_2 到 P_6 的转移概率是最高的。

还应注意的是，某些页面的转移概率可能是不可用的。例如，从 P_2 到 P_5 的转移概率不可用，因为没有用户在 P_2 之后访问过 P_5。因此，这些转移概率设置为零。类似地，K 阶马尔可夫模型是在考虑了用户执行的最后 K 次动作之后，计算预测的式（B.13）。在 WWW 预测中，第 K 阶马尔可夫模型是考虑到其之前的 $k-1$ 页访问时，用户访问第 k 页面的概率。

图 B.10 所示为对应于表 B.1 的二阶马尔可夫模型。在二阶模型中，我们考虑最后 2 页。转移概率以类似的方式计算。例如，由于用户从状态（P_1，P_2）到状态（P_2，P_6）的次数为 1，转移（P_1，P_2）到（P_2，P_6）的转移概率为 0.16（即，0.16 = 1 × 1/6）。转移概率用于预测。例如，考虑到用户访问了 P_1 和 P_2，接下来他/她最有可能访问 P_4，因为从状态（P_1，P_2）到状态（P_2，P_4）的转移概率大于从状态（P_1，P_2）到状态（P_2，P_6）的转移概率。

马尔可夫模型的阶数与滑动窗口有关。K 阶马尔可夫模型对应于大小为 $K-1$ 的滑动窗口。

应注意到，还有另一个概念类似于滑动窗口概念，即跳数。在本论文中，我们可以互换使用跳数和滑动窗口。

在 WWW 预测中，马尔可夫模型是基于 n－gram 的概念而构建的。n－gram 可以表

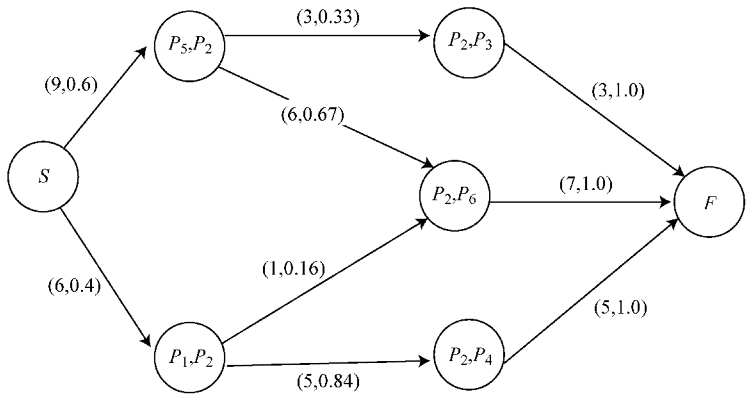


图 B.10　二阶马尔可夫模型

示为 $<x_1, x_2, \cdots, x_n>$ 形式的元组，以描绘用户浏览网站页面的点击次数。n－gram 的每个组件都具有特定的页面 ID 值，该值反映了特定用户浏览网页的路径。例如，某些用户 U 的 n－gram $<P_{10}, P_{21}, P_4, P_{12}>$ 表示用户 U 已经以序列访问了页面 10，21、4 和最后的页面 12。

B.6　关联规则挖掘

关联规则是一种数据挖掘技术，已成功应用于发掘相关事务。关联规则技术根据项目集在不同事务中能同时出现，来找到项目集之间的关系。具体来说，ARM 会发现这些项目集中的频繁模式（规律性）。例如，超级商店一起购买的商品是什么？下面简要介绍一下 ARM。有关更多细节，请参见（本附录参考文献［AGRA93］、［AGRA94］）。

假设我们的数据库中有 m 个项目，将 $I=\{i_1, i_2, \cdots, i_m\}$ 定义为所有项目的集合。事务 T 是一组项目，使得 $T \subseteq I$，令 D 是数据库中所有事务的集合。事务 T 包含 X，如果 $X \subseteq T$ 和 $X \subseteq I$，关联规则就是 $X \rightarrow Y$ 形式的含义，其中，$X \subset I$，$Y \subset I$ 和 $X \cap Y = \phi$。考虑规则有两个参数：置信度和支持度。如果 D 中 $c\%$ 的事务包含 Y［即，$c = pr(Y \mid X)$］，则规则 $R = X \rightarrow Y$ 有置信度 c。如果 D 中 $s\%$ 的事务包含 X 和 Y［即 $s = pr(X, Y)$］，则规则 R 有置信度 s。挖掘关联规则的问题定义为：给定一组事务 D，我们希望生成满足置信度和大于最小置信度（σ）以及最小支持度 minsup（θ）的所有规则。有几种有效的算法可以找到关联规则。比如 AIS 算法（见本附录参考文献［AGRA93］、［AGRA94］）、SETM 算法（见本附录参考文献［HOUT95］）和 AprioriTid（见本附录参考文献［AGRA94］）等关联规则。

在 Web 事务的情况下，我们使用关联规则来发现用户之间的网页导航模式。这将有助于提前缓存页面，并减少页面的加载时间。此外，发现网页导航模式有助于个性化。事务是从 Web 服务器日志中获取被捕捉到的单击流数据。

在许多应用中，使用 ARM 有一个主要问题。首先，使用全局最小支持度（minsup）

的问题是，由于极少的单击量，即很少访问的网页，其不会被包含在频繁项目集中，因为它不能获得足够的支持。一个解决方案是设定一个非常小的支持度阈值，然而，这样的话我们最终会得到一个非常大的频繁项目集，这在计算上是难以处理的。刘等人（见本附录参考文献［LIU99］）讨论了对不同项目使用不同支持阈值的挖掘技术。指定多个阈值允许将罕见的事务（可能非常重要）包含在频繁项目集中。其他问题可能会随应用程序本身的运行而出现。例如，在 WWW 预测的情况下，为每个用户记录会话。会话可能有几十个单击流（有时候为数百个会话的持续时间）。将每个会话用作单一事务将不起作用，因为很少发现频繁重复（即相同）的两个会话。因此，它不会达到甚至非常高的支持度阈值（minsup）。这需要将每个会话分成许多子序列。一种常见的方法是使用尺寸为 w 的滑动窗口。例如，假设我们使用滑动窗口 $w=3$ 来中断会话 $S=<A,B,C,D,E,F>$，则我们将得到子序列 $S'=\{<A,B,C>, <B,C,D>, <C,D,E>, <D,E,F>\}$。使用窗口 w 的会话 S 的子序列的总数是长度$(S)-w$。为了在用户会话中预测活动的下一页，我们使用活动会话的滑动窗口，并忽略以前的页面。例如，如果当前会话是 $<A,B,C>$，并且用户将引用页面 D，则新的活动会话将使用滑动窗口 3 变为 $<B,C,D>$。应注意到，页面 A 被删除，且 $<B,C,D>$ 将用于预测。这样做的理由是因为大多数用户在浏览网页时，只是试图找到所需的信息，而这种使用用户浏览历史中的最近部分来生成建议/预测的做法，是最适合这种情况的（见本附录参考文献［MOBA01］）。

Mobasher 等人（见本附录参考文献［MOBA01］）讨论了一个将活动用户会话与数据库中的频繁项目集相匹配的推荐引擎，并预测用户最可能访问的下一页。引擎的工作原理如下。给定大小为 w 的活动会话，引擎查找长度为 $w+1$ 的所有频繁项目集，其满足一些最小支持度（minsup）并包含当前活动会话。活动会话 A 的预测基于相应关联规则的置信度（ψ）。关联规则 $X\to z$ 的置信度（ψ）定义为 $\psi(X\to z)=\sigma(X\cup z)/\sigma(X)$，其中 z 的长度为 1。第 p 页被推荐/预测为活动会话 A，当且仅当

$\forall V$，R 中的频繁项目集

$$\text{length}(R)=\text{length}(V)=\text{length}(A)+1\wedge$$

$$R=A\cup\{p\}\wedge$$

$$V=A\cup\{q\}\wedge$$

$$\psi(A\to p)>\psi(A\to q)$$

引擎使用一个称为频繁项目集图的循环图。该图是本附录参考文献［AGRA01］的树投影算法中所使用的词典树的扩展。图形按层次组织。级别 1 中的节点具有大小为 1 的项目集。例如，级别 1 和 2 中节点的大小（即，对应于这些节点的项目集的大小）分别为 1 和 2。图的根，级别 0 是对应于空项目集的空节点。如果 $X\subset Y$，则级别 l 中的节点 X 链接到级别 $l+1$ 中的节点 Y。为了进一步解释该过程，假设我们具有以下示例 Web 事务，涉及第 1、2、3、4 和 5 页，如表 B.2 所示。先验算法使用 minsup = 0.49 生成表 B.3 中的项目集。频繁项目集图如图 B.11 所示。

假设我们使用大小为 2 的滑动窗口，当前活动会话 $A=<2,3>$。为了预测/推荐

下一页，我们首先从频繁项目集图中的第2级开始，并提取链接到A的3级中的所有项目集。从图B.11，节点{2，3}链接到{1，2，3}和{2，3，5}节点：

$$\psi(\{2,3\}\rightarrow 1)=\sigma(\{1,2,3\}/\sigma\{2,3\})=5/5=1.0$$

$$\psi(\{2,3\}\rightarrow 5)=\sigma(\{2,3,5\}/\sigma\{2,3\})=4/5=0.8$$

表 B.2 Web 事务示例

事务 ID	项目
T1	1，2，4，5
T2	1，2，5，3，4
T3	1，2，5，3
T4	2，5，2，1，3
T5	4，1，2，5，3
T6	1，2，3，4
T7	4，5
T8	4，5，3，1

表 B.3 由 Priori 算法生成的频繁项集

尺寸1	尺寸2	尺寸3	尺寸4
{2}（6）	{2，3}（5）	{2，3，1}（5）	{2，3，1，5}（4）
{3}（6）	{2，4}（4）	{2，3，5}（4）	
{4}（6）	{2，1}（6）	{2，4，1}（4）	
{1}（7）	{2，5}（5）	{2，1，5}（5）	
{5}（7）	{3，4}（4）	{3，4，1}（4）	
	{3，1}（6）	{3，1，5}（5）	
	{3，5}（5）	{4，1，5}（4）	
	{4，1}（5）		
	{4，5}（5）		
	{1，5}（6）		

推荐的页面是1，因为它的置信度更大。需要注意的是，在推荐引擎中，不考虑单击流的顺序，也就是说，会话<1，2，4>和<1，4，2>之间没有区别。这是该系统的缺点，因为访问页面的顺序可能包含关于用户网页导航模式的重要信息。

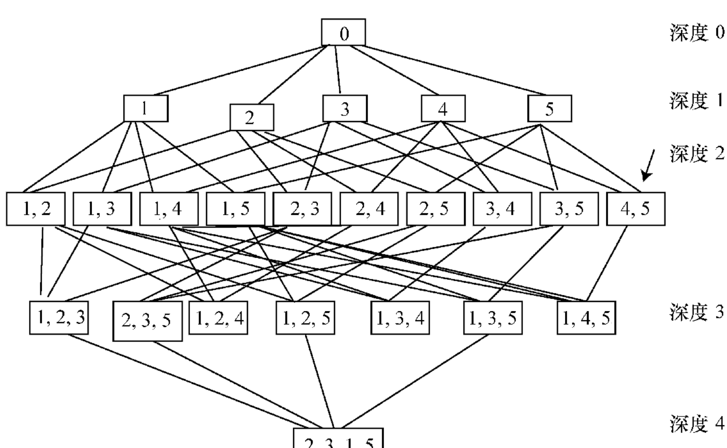


图 B.11 频繁项目集曲线图

B.7 多类问题

大多数分类技术用以解决二分类问题。二分类技术叠加使用，可以解决多分类问题。这种泛化有两个基本方案，即一对一和一对全部。为了避免冗余，我们将只给出SVM 的泛化。

B.7.1 一对一

一对一方案为每对类创建一个分类器。每对分类器（i，j）的训练集仅包括属于 i 类或 j 类的那些实例。一个新的实例 x 属于大多数分类器对认可的那一类。预测决定来自于多数投票技术。若 n 是数据集中的类数，那么就有 n（$n-1$）/ 2 个分类器要计算。显然，该方案的缺点是我们需要生成大量的分类器，特别是如果训练集中有大量的类。例如，如果我们有一个 1000 个类的训练集，我们需要 499500 个分类器。另一方面，每个分类器的训练集很小，因为我们要排除不属于该对类的所有实例。

B.7.2 一对全部

一对全部方案为数据集中的每个类创建一个分类器。训练集被预处理，使得对于分类器 j 属于类 j 的实例被标记为类（+1），不属于类 j 的实例被标记为类（-1）。在一对全部方案中，我们计算 n 个分类器，其中 n 是用户访问的页面数（在每个会话结束时）。通过将新的实例 x 分配给类，其分类器输出最大正值（即最大边际），如式（B.15）所示。我们可以计算点 x 的边界，如式（B.14）所示。需注意的是，推荐/预测页面是该页面中边距值的符号［参见式（B.10）］。

$$f(x) = wx - b = \sum_{i}^{N} \alpha_i y_i (x \cdot x_i - b) \tag{B.14}$$

$$\text{prediction}(x) = \text{argmax}_{1 \leq c \leq M} f_c(x) \tag{B.15}$$

在式（B.15）中，M 是类的数量，$x = < x_1, x_2, \cdots, x_n >$ 是用户会话，f_i 是将类 i 与其余类分隔的分类器。式（B.15）中的预测决定说明了，与测试示例 x 最远的分类器为 f_c。这可能被解释为 f_c 具有最大的分离能力，在所有其他分类器中，将 x 与其余类分离。

与一对一方案相比，该方案（一对一全部）的优点在于它具有较少的分类器。在另一方面，训练集的大小方面是一对全部比一对一的方案大，因为我们要使用整个原始训练集来计算每个分类。

B.8 图像挖掘

随着数字图像和计算机存储技术的发展，每天都会生成并保存大量的数字图像。数字图像的应用迅速渗透到许多领域和市场，包括商业和新闻媒体照片库、科学和非照相图像数据库，以及医学图像数据库。因此，我们面临着一个令人畏缩的问题，即要组织和访问这些海量可用图像。人们非常希望一个有效的图像检索系统能从数据库中查找特定实体的图像。该系统可以有效地管理大量图像，高速响应用户的查询，提供最少的无

关信息（高精度），并确保相关信息不被忽视（高回调）。

为了做出这样的系统，人们尝试了许多不同的方法。在 20 世纪 90 年代初，由于出现了大量的图像集合，人们讨论了基于内容的图像检索（CBIR）。CBIR 基于视觉内容/低级图像特征（如颜色直方图、纹理、形状和空间布局）的相似性来计算相关性。然而，问题是视觉相似性不是语义相似性。低级视觉特征与语义含义之间存在差距。所谓的语义差距是大多数 CBIR 方法需要解决的主要问题。例如，CBIR 系统可以用“红玫瑰”的图像来回答对“红球”的查询请求。如果我们用关键字对图像进行注释，则发布图像数据存储库的典型方法就是创建一个基于关键字的查询接口寻址到图像数据库。如果所有图像都具有详细和准确的描述，则基于当前强大的纯文本搜索技术，图像检索将是方便的。如果这些搜索技术的描述/注释包含用户指定的关键字的某些组合，那么它们将检索图像。然而，主要的问题是大多数图像没有注释。手动注释大量图像是一个费力、且容易出错的主观过程，因为即使它们不包含用户指定的关键字，但许多图像要包含所需的语义信息。此外，基于关键字的搜索是有用的，尤其对于知道什么样的关键字用于索引图像的用户，因此他们很容易制定出查询语句。然而，当用户没有明确的目标时，不知道数据库中是什么，并且不知道域中涉及什么样的语义概念时，这种方法是有问题的。

图像挖掘是比 CBIR 系统中检索相关图像更具挑战性的研究问题。图像挖掘的目标是找到一个图像模式，这对于给定的一组图像是很重要的，这有助于了解高级语义概念/语义描述与低级视觉特征之间的关系。我们阐述的重点是特征选择和图像分类等方面。

B. 8. 1　特征选择

通常，保存在数据库中的数据具有明确定义的语义。例如，数字或结构化数据条目。相比之下，具有不完整定义语义的数据是非结构化数据。例如，图像、音频和视频是具有不明确语义的数据。在图像处理领域，图像由导出的数据或特征（如颜色、纹理和形状）表示。这些特征中的许多具有多个值（例如，颜色直方图、时间描述）。当人们生成这些导出的数据时，它们通常就会产生尽可能多的特征，因为它们不知道哪个特征更相关。因此，导出图像数据的维数通常很高。实际上，某些所选特征可能会重复，甚至可能与该问题无关。被包含的不相关或重复的信息被称为噪声。这些问题被称为“维度诅咒”。特征选择是寻找特征中最佳子集的研究课题。在本论文中，我们将详细讨论这种诅咒和特征选择。

我们开发了一种基于包装器的同步特征加权和聚类算法。聚类算法将相似的图像段捆绑在一起，并生成有限的视觉符号集（即，blob - token）。在直方图分析和卡方值的基础上，我们对图像段的特征分配不同的权重，而不是去除其中的一些。特征权重评估被包裹在聚类算法中。在算法的每次迭代中，基于聚类结果重新评估图像段的特征权重。重新评估的特征权重将影响下一次迭代中的聚类结果。

B. 8. 2　自动图像注释

自动图像注释是关于对象识别的研究，其中涉及尝试识别图像中的对象，并且根据

对象的语义生成图像的描述。如果可以为图像生成准确和完整的语义描述，那么我们就可以在图像数据库中存储该描述。基于文本描述，通过应用许多现有的基于文本的搜索技术，可以容易且有效地实现图像 DBMS 的更多功能（例如浏览、搜索和查询）。不过，自动图像注释问题目前还没有解决，也许这个问题是不可能解决的。

然而，在某些子域中，仍然可以获得一些有趣的结果。许多统计模型已经发布并用于图像注释。这些模型中的一些实例已经考虑了特征维度，并应用奇异值分解（Singular – Value Decomposition，SVD）或主成分分析（Principle Component Analysis，PCA）来减小维数。但他们没有考虑特征选择或特征权重。我们基于翻译模型（Translation Model，TM）开发了一个用于图像注释的新框架。在我们的方法中，我们应用了加权特征选择算法并将其嵌入到图像注释框架中。我们的加权特征选择算法提高了视觉令牌的质量，并能生成更好的图像注释。

B. 8. 3 图像分类

图像分类是一个重要的领域，特别是在医疗领域。因为它有助于管理大型医学图像数据库，并且在现实临床环境中具有很大的诊断辅助潜力。下面描述我们的 CLEF 图像检索任务的实验。CLEF 医学图像数据集的大小不均衡，这对所有分类算法来说都是一个非常严重的问题。为了解决这个问题，我们通过生成子窗口重新采样数据。K 最近邻（K Nearest Neighbor，KNN）算法、距离加权 KNN、模糊 KNN、最近的原型分类器和基于证据理论的 KNN 正在被研究。结果表明，基于证据的 KNN 具有基于分类精度的最佳性能。

B. 9 总结和展望

在本附录中，我们首先提供了各种数据挖掘任务和技术的概述，然后讨论了我们在本书中使用的一些技术。这些包括神经网络、SVM 和 ARM。众多的数据挖掘技术已被设计和开发，其中许多技术都被用作商业工具。其中几个技术是一些基本分类、聚类和 ARM 技术的变体。目前主要的挑战之一是为各种应用选择合适的技术。在这方面，我们仍然需要更多的基准测试和性能研究。此外，选择的技术应有更少的误报和漏报。虽然未来仍有许多工作要做，但在过去 10 年的进步也是非常值得肯定的。

参 考 文 献

AGRA93] Agrawal, R., T. Imielinski, and A. Swami, Mining association rules between sets of items in large database. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Washington, DC, May 1993, pp. 207–216.

AGRA94] Agrawal, R. and R. Srikant, Fast algorithms for mining association rules in large database. In *Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, CA, 1994, pp. 487–499.

AGRA01] Agrawal, R., C. Aggarawal, and V. Prasad, A tree projection algorithm for generation of frequent item sets. *Journal of Parallel and Distributed Computing Archive*, 61(3): 350–371, 2001, Orlando, FL.

BART99] Bartlett, P. and J. Shawe-Taylor, Generalization performance of support vector machines and other pattern classifiers. In: (B. Schölkopf, C. Burges, A. Smola, Eds.)

Advances in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, UK, 1999, pp. 43–54.
[CRIS00] Cristianini, N. and J. Shawe-Taylor, *Introduction to Support Vector Machines*, 1st ed., Cambridge University Press, 2000, pp. 93–122, Cambridge, UK.
[HOUT95] Houtsma, M. and A. Swanu, Set-oriented mining of association rules in relational databases. In *Proceedings of the Eleventh International Conference on Data Engineering*, Washington, DC, 1995, pp. 25–33.
[LIU99] Liu, B., W. Hsu, and Y. Ma, Association rules with multiple minimum supports. in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge discovery and Data Mining*, San Diego, CA, 1999, pp. 337–341.
[MITC97] Mitchell, T.M, *Machine Learning*, McGraw-Hill, 1997, chap. 4.
[MOBA01] Mobasher, B., H. Dai, T. Luo, and M. Nakagawa, Effective personalization based on association rule discovery from web usage data. In *Proceedings of the ACM Workshop on Web Information and Data Management (WIDM01)*, Atlanda, GA, 2001, pp. 9–15.
[PIRO96] Pirolli, P., J. Pitkow, and R. Rao, Silk from a sow's ear: Extracting usable structures from the web. In *Proceedings of 1996 Conference on Human Factors in Computing Systems (CHI-96)*, Vancouver, British Columbia, Canada, 1996, pp. 118–125.
[VAPN95] Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer, New York, NY, 1995.
[VAPN98] Vapnik, V.N., *Statistical Learning Theory*, Wiley, New York, 1998.
[VAPN99] Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, NY, 1999.
[YANG01] Yang, Q., H. Zhang, and T. Li, Mining web logs for prediction models in WWW caching and prefetching. In *The 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD*, San Francisco, CA, August 26–29, 2001, pp. 473–478.

附录 C　数据库系统中的访问控制

C.1　概述

由于本书中大部分内容涉及了云数据管理和云数据安全管理，因此我们将对数据管理系统的访问控制提供比较全面的概述。特别是，我们要讨论安全策略以及执行数据库系统中的策略。我们将重点关注放在自主安全策略上，有关数据安全管理的更多详细信息，请参见本附录参考文献［FERR00］和［THUR05］。

最流行的自主安全策略是访问控制策略。20 世纪 60 年代人们对操作系统的访问控制策略进行了研究，然后在 20 世纪 70 年代人们对数据库系统进行了研究。两个著名的数据库系统 System R 和 INGRES 是调查数据库系统访问控制的首选之一（参见本附录参考文献［GRIF76］和［STON74］）。此后已经报告了访问控制策略的几个变体，以及其他自主策略（包括管理策略）。我们还在自主策略下讨论识别和认证。需要注意的是，本附录中的大部分讨论将侧重于关系数据库系统中的自主安全性。许多原则可以应用于其他系统，如对象数据库系统、分布式数据库系统和云数据管理系统。

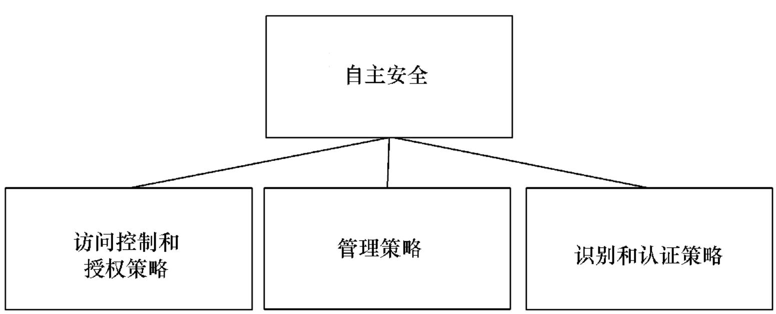
在设计安全系统之前，必须回答的第一个问题是，系统要执行的安全策略是什么？安全策略本质上是一套强制执行安全性的规则。安全策略包括强制安全策略和自主安全策略。强制安全策略是“强制性”的策略，不应被忽视。自主安全策略是由管理员或负责系统运行环境的任何人所指定的策略。

策略执行，我们是指执行策略的机制。例如，早在20世纪70年代，像System R和INGRES等关系数据库系统产品，开发了如用于策略执行的查询修改机制的技术（参见本附录参考文献［GRIF76］和［STON74］）。现在，查询语言SQL已被扩展，以指定安全策略和访问控制规则。最近，已经扩展了如XML和RDF的语言用来指定安全策略（参见本附录参考文献［BERT02］和［CARM04］）。

在C.2节中，我们介绍自主安全性，包括数据库系统的访问控制和授权模型。我们还要讨论基于角色的访问控制系统。在C.3节中，我们讨论执行自主安全性的方法，包括对查询修改的讨论。我们还提供各种商业产品的概述。第C.4节总结本附录。需要说明的是，本附录中的论述将主要概述基于关系数据库系统中自主安全的基础知识。

C.2 安全策略

本节的组织如下：在C.2.1节中，我们将对访问控制策略进行概述。管理策略将在C.2.2节中讨论。关于识别和认证的问题将在C.2.3节中进行讨论。C.2.4节将讨论审计数据库管理系统。作为安全对象的视图将在C.2.5节中讨论。图C.1所示为自主安全策略的各个组成部分。



图C.1 自主安全策略

C.2.1 访问控制策略

首先检查操作系统的访问控制策略。这里的要点是可以允许进程访问文件吗？访问可以是进行读访问或写访问。写访问包括允许修改、添加或删除。这些原则被移植到如INGRES和System R等数据库系统中。从那时起，我们已经研究了各种形式的访问控制策略。其中值得注意的是，在几个商业系统中，基于角色的访问控制策略目前已经得以实施。还应注意到，访问控制策略还包括强制性策略。图C.2所示为各种访问控制策略。

C.2.1.1 基于授权的访问控制策略

许多访问控制策略基于授权策略。基本上这意味着用户被授予对数据访问的权限，

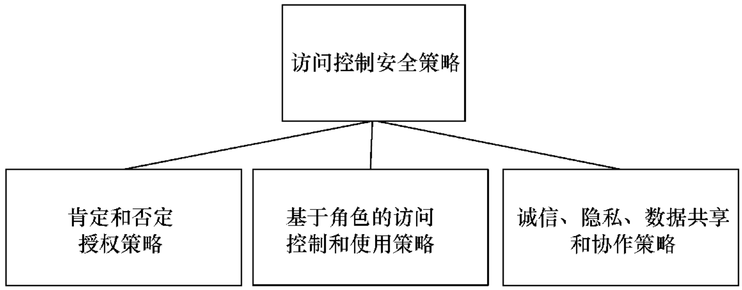


图 C.2　访问控制安全策略

是基于授权规则的。在本节中，我们将讨论各种授权规则。需要注意的是，授权策略在 Ferrari 和 Turaisingham 的书中有详细的讨论（见本附录参考文献［FERR00］）。

肯定授权：早期的系统重点关注的是现在被称为肯定授权的规则。这里，用户 John 被授予访问关系 EMP 或用户 Jane 被授予访问关系 DEPT。这些就是关系的访问控制规则。还可以授予对其他实体（如属性和元组）的访问权限。例如，John 已经有在 EMP 相关域中读取属性“工资”和写入属性“名字”的权限。写访问可以包括增加、修改或删除。

否定授权：问题是如果没有指定 John 对一个对象的访问权限，这是否意味着 John 无法访问该对象？在某些系统中，没有指定的任何授权规则被隐含地视为否定授权，而在其他系统中，就会明确指定了否定授权。例如，我们可以执行如 John 无法访问关系 EMP，或 Jane 无法访问关系 DEPT 的规则。

冲突消除：当我们有规则存在冲突的时候，那么我们将如何解决这个冲突？例如，我们可以有一个规则来授予 John 对关系 EMP 的访问权限。然而，我们也可以有另一个规则，不授予 John 在 EMP 中读取工资属性的权限。这就是一个冲突。通常，系统执行最小特权规则，而在这种情况下，John 可以访问关系 EMP，除了工资值。

强弱授权：系统还强制执行强弱授权。在强授权的情况下，无论发生什么冲突，规则都成立。在弱授权的情况下，其规则在发生冲突时不适用。例如，如果 John 被授予对 EMP 的访问权限，并且它是一个强授权规则，并且没有授予给 John 访问工资属性的规则是弱授权，此时就存在一个冲突。而这意味着强授权将成立。

授权规则的传播：这里的问题是规则将如何传播？例如，如果 John 已经有读取关于 EMP 关系的访问权限，那么它是否意味着 John 已经自动有读取 EMP 关系中每个元素的访问权限？通常情况下，除非我们有一个规则来禁止自动传播授权规则。如果我们有一个规则来禁止规则自动传播，那么我们就必须明确地强制执行授权规则，该规则指定 John 可以访问的对象。

特殊规则：在强制性策略工作中，我们广泛探讨了内容和基于上下文的约束的实施。需要注意的是，安全约束本质上是安全规则。内容和基于上下文的规则是根据数据的内容或其中显示数据的上下文来授予访问权限的规则。这种规则也可以用于自主安全从而被执行。例如，在基于内容的约束情况下，John 只能在 DEPT D100 中有读取元组的访问权限。在上下文或基于关联的约束情况下，John 没有合在一起读取名字和工资

的访问权限，但是，他可以单独地访问名字和工资。在基于事件的约束情况下，执行选择之后，John 就可以访问 EMP 关系中的所有元素。

规则的一致性和完整性：这里的一个挑战是确保约束的一致性和完整性。也就是说，如果约束或规则不一致，那么我们是否有解决冲突的办法呢？我们如何确保在用户的访问控制规则中指定所有实体（如属性、关系、元素等）？实际上这意味着，规则是完整的吗？如果没有对特定用户或一类用户指定肯定或否定授权的实体，那么我们将对其假定什么呢？

我们已经就授权规则讨论了一些要点。图 C.3 给出了一些例子。在下一节中，我们将讨论一个非常受欢迎的访问控制策略，这就是基于角色的访问控制，现已经在商业系统中得到了实施。

授权规则

- Joh拥有对员工关系的读取访问权限
- Joh不拥有对部门相关域的写访问权限
- 在员工关系中Jane拥有对名字值的读取访问权限
- Jane不拥有读取访问部门关系的访问权限

图 C.3　授权规则

C.2.1.2　基于角色的访问控制

基于角色的访问控制已经成为最受欢迎的访问控制方法之一（参见本附录参考文献［SAND96］）。这种方法已在包括 Trusted Oracle 在内的商业系统中实现。这里的思想是根据用户的角色和职能授予用户访问权限。

基于角色的访问控制的基本思想也称为 RBAC，具体如下：用户需要根据其角色访问数据。例如，董事长可以获得关于他/她的副董事长和董事会成员的信息，而财务总监可能会获得财务信息和关于向他报告的人的信息。董事可以获取关于在其部门工作的人员的信息，而人力资源总监将获得关于公司员工个人资料的信息。基本上基于角色的访问控制是一种授权策略，它取决于用户角色以及与角色相关的活动。

相关文献中已经讨论了各种关于角色层次结构的研究成果。还有一个名为 SACMAT（访问控制模型和技术研讨会，Symposium on Access Control Models and Technologies）的会议系列，它从基于角色的访问控制研究工作中演变而来。例如，访问如何得到传播？一个角色能否包含另一个角色？考虑图 C.4 中所示的角色层次。这意味着如果我们允许访问层次结构中的节点，那么访问是否向上传播？也就是说，如果部门经理有权访问某些项目信息，那么该访问是否传播到父节点（它是主管节点）？如果部门负责人可以访问他/她的部门中的员工信息，该访问是否传播给作为角色层次结构中的上一级的部门经理？子节点会发生什么？也就是说，访问是否向下传播？例如，如果部门经理可以访问某些信息，那么他的下属可以访问该信息吗？是否存在下属可以访问部门经理没有访问的数据情况？如果员工必须向两位主管报告，一

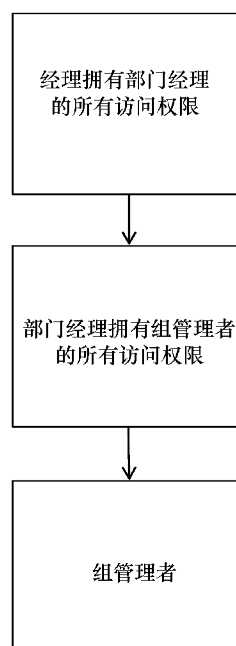


图 C.4　角色层次

位是他的部门经理，而另一位是他的项目经理，该怎么办？当部门经理正在开展项目，且还要向他的项目负责人报告，那么会发生什么情况？

我们已经对关系系统、对象系统、分布式系统以及现在的一些新兴技术，如数据仓库、知识管理系统、语义 Web、电子商务系统和数字图书馆进行了基于角色访问控制的应用研究。此外，对象模型已用于表示角色和活动（可以参见 IFIP 数据库安全会议系列的会议记录）。这是一个将持续讨论的领域，ACM SACMAT 是发布关于这个主题的高质量论文的场所。

最近，Sandhu 等人开发了另一种类似的访问控制模型，这就是使用控制模型，他称之为 UCON（参见本附录参考文献［PARK04］中报道的工作）。UCON 模型试图整合 3 种策略，它们是信任管理、访问控制和权限管理。这个想法是提供对使用对象的控制。虽然这些想法仍处于初步阶段，但我们对这种模型的发展抱有很大的希望。

C. 2. 2　管理策略

当访问控制策略指定用户对数据有用特定的访问权限时，管理策略将指定谁来管理数据。管理职责包括保持数据流动，确保在更新数据时更新元数据，并确保从故障和相关活动中恢复。

通常，数据库管理员（Database Administer，DBA）负责更新数据，比如元数据、索引和访问方法，并确保访问控制规则被正确执行。系统安全员（System Security Officer，SSO）也具有属于自己的角色。也就是说，DBA 和 SSO 可以分担他们之间的职责。与安全相关的问题可能是 SSO 的责任，而与数据相关的问题可能是 DBA 的责任。我们正在考虑一些其他的管理策略（包括分配管理员）。通常所有者可以控制他们创建的数据，并可以管理数据的持续时间。在某些情况下，系统拥有者可能无法管理数据，在这种情况下，他们可能就会分配管理员来管理。

分布式环境中的管理策略变得更加复杂，特别是在 Web 环境中。例如，在 Web 环境中，分发文档可能涉及多方，包括所有者、发布者和请求数据的用户。谁拥有数据，是所有者还是发布者？一旦数据离开了所有者后到达发布者，发布者是否能够控制数据？

当我们从关系数据库环境迁移到分布式且可能是网络环境时，会有许多有意义的问题需要来回答。这些包括管理版权问题、数据质量、数据来源和数据管理。最近在管理策略会议上出现了许多有趣的论文。图 C. 5 所示为各种管理策略。

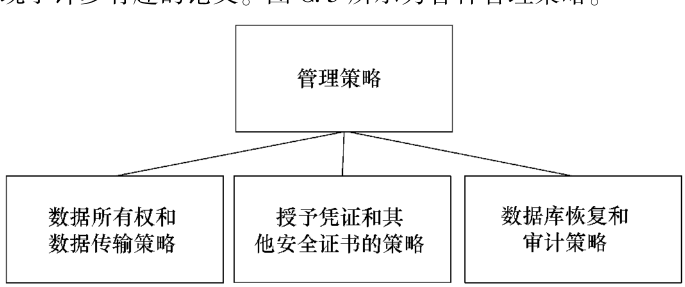


图 C. 5　管理策略

C.2.3 识别和认证

作为我们对自主安全讨论中的一部分，我们还要讨论识别和认证。通过识别，我们的意思是用户必须用自己的用户 ID 和密码来标识自己。认证意味着系统必须将用户 ID 与密码相匹配，以确保这确实是他本人所声明的。取决于用户的角色，他/她也可能具有多个身份。身份管理最近引起了广泛关注（参见本附录参考文献［BERT02］）。

基于密码的方案报出了许多问题。其中一个问题是黑客可以进入系统并获取用户的密码，然后伪装成用户。在集中式系统中，问题并不像分布式环境那么复杂。现在，随着万维网和电子商务的应用，当黑客伪装成合法用户时，金融机构将可能会损失数十亿美元。

最近，正在应用生物识别技术。这些包括用于认证用户的面部识别和语音识别技术。这些已经被使用的技术进一步提升了我们的安全性。随着人脸识别技术的进步，我们可以期待生物识别技术的广泛使用。

C.2.4 审计数据库系统

出于多个目的，多个数据需要被审计。例如，可以对它们进行审计，以跟踪所提出的查询数量、所做更新的次数、执行的事务数以及辅助存储器被访问的次数，以便系统可以被设计得更有效率。为了安全起见，数据库也可以被审计。例如，是否通过向用户发布信息来绕过任何访问控制规则？推理问题是否发生？是否有侵犯隐私吗？是否有未经授权的入侵？

审计会创建一个跟踪记录，并将审计数据存储在数据库中。该数据库可能被挖掘以检测任何异常模式或行为。在使用数据挖掘进行审计和入侵检测方面已经做了很多工作。这些日子，审计跟踪分析在网络上的电子商务交易方面尤其重要。机构应有能力进行分析、确定信用卡欺诈和身份盗用等问题。

C.2.5 安全视图

为了实现自主安全和强制安全，视图作为一项安全机制已经被大量研究。例如，人们可能不想授予对整个关系的访问。特别是，假如它已经声明 25 个属性，如医疗保健记录、工资、旅行信息、个人数据等。因此，DBA 可以形成视图并授予访问权限。同样在强制性安全的情况下，视图可以分配安全级别，我们在第四部分中讨论这一点。

视图也有与它们相关的问题，它们包括视图更新问题（见本附录参考文献［DATE90］）。也就是说，如果视图被更新，那么我们就需要确保完成更新基础关系。因此，如果用户 John 更新了一个视图，那么他无法访问基本相关域，那么基本关系是否仍然可以更新？也就是说，我们是否为不同的用户创建不同的视图，然后 DBA 将视图的更新合并为基础关系的更新？图 C.6 所示为安全性的视图。

C.3 策略实施及相关问题

本节的组织如下。C.3.1 节讨论用于安全的 SQL 扩展。在 C.3.2 节，我们讨论查询修改。C.3.3 节将讨论自主安全对其他数据库功能的影响。C.3.4 节讨论安全策略的可视化。C.3.5 节将讨论实施自主安全策略的原型和产品。需要注意的是，我们将重点关

EMP			
SS#	Ename	Salary	D#
1 1	John	20k	10
2 2	Paul	30k	20
3 3	Mary	40k	20
4 4	Jane	20k	20
5 5	Bill	20k	10
6 6	Larry	20k	10
1 1	Michelle	30k	20

V1.VIEW EMP(D#=20)		
SS#	Ename	Salary
2	Paul	30k
3	Mary	40k
4	Jane	20k
1	Michelle	30k

V2.VIEW EMP(D#=10)		
SS#	Ename	Salary
1	John	20k
5	Bill	20k
6	Larry	20k

规则
John有读取V1的权限
John有对V2进行写入的访问

图 C.6　安全视图

注关系数据库系统。图 C.7 所示为执行安全策略所涉及的各个方面，包括规范、实现和可视化。

策略实施

查询修改算法

执行访问控制的规则处理

定理证明技术是否违反政策

策略的一致性和完整性检查

图 C.7　策略实施

C.3.1　为了安全的 SQL 扩展

本节讨论策略规范。虽然大部分重点将放在针对安全策略规范的 SQL 扩展上，但我们还将讨论一些新兴语言。应注意到，SQL 是为关系系统的数据定义和数据操作而开发的。已经开发了各种版本的 SQL，包括面向对象的 SQL、面向多媒体的 SQL 和面向 Web 的 SQL。也就是说，SQL 在过去 20 年中已经大大地影响了数据操作和数据定义（见本附录参考文献［SQL3］）。

正如我们所说，SQL 是一种数据定义和数据操作语言。在数据定义期间可以指定安全策略。SQL 具有 GRANT 和 REVOKE 结构，用于指定授权和撤销对用户的访问。也就是说，如果用户 John 已经有读取关系 EMP 的访问权限，那么就可以使用 SQL 并指定类似“GRANT JOHN EMP READ”的内容，如果要撤销访问权限，那么我们就需要一些类似“REVOKE JOHN EMP READ”的内容。“SQL 也扩展了更复杂的约束，如授予 John 对相关域中元组的访问权限，并授予 Jane 对相关域中元素的访问权限。

在本附录参考文献［THUR89］中，我们为安全性声明指定了 SQL 扩展。他们的声明是为了多层次的安全。我们可以使用类似的推理来指定自主安全策略。例如，考虑到

John 没有 EMP 关系中名字和工资属性组合的读访问权限情况，但他可以单独读取名字和工资。人们可以用类似 SQL 的语言来指定它，如下所示。

GRANT JOHN READ
EMP.SALARY
GRANT JOHN READ
EMP.NAME
NOT GRANT JOHN READ
Together (EMP.NAME, EMP.SALARY)

如果我们要授予 John 读取工资低于 30k 的员工的访问权限，那么这个声明指定如下。

GRANT JOHN READ
EMP
Where EMP.SALARY < 30 K

需要注意的是，我们指定的声明不是标准声明。这些也只是我们的部分想法。我们需要探索将这些声明纳入标准的方法。我们还提出了 SQL 扩展，用于基于角色的访问控制。实际上，像 Oracle 的 Trusted 数据库这样的产品就可以强制实施基于角色的访问控制。访问控制规则以类似 SQL 的语言指定。也应注意到，还有许多其他规范语言也已经被开发出来，包括 XML、RDF 以及 Web 和语义 Web 的相关语言。图 C. 8 所示为安全策略包含的各种规范。

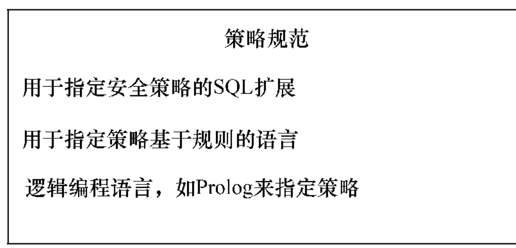


图 C. 8　策略规范

C. 3. 2　查询修改

查询修改在加州大学伯克利分校的 INGRES 项目中被首次提出（见本附录参考文献［STON74］）。其思想是根据约束修改查询。我们已经成功地设计并实施了强制性安全的查询修改（参见本附录参考文献［DWYE87］、［THUR87］、［THUR93］）。然而，本节中的大部分讨论将是基于自主安全约束的查询修改。我们用一些例子说明核心内容。

考虑用户 John 的查询操作，以从 EMP 中检索所有元组。假设 John 只读取所有元组中工资低于 30k 的员工的访问权限并且该员工不在安全部门。

那么查询

Select * from EMP

查询将会被改为

Select * from EMP
Where salary < 30 k
And Dept is not Security

我们假设 EMP 的属性是姓名字、工资、年龄和部门。

这里，实际上执行的是查询“where”子句关联这个关系的所有约束。我们也可以

有跨越多个关系的约束。例如，Dept#可以加入 EMP 和 DEPT 两个相关域。然后查询如下修改：

```
Select * from EMP
Where EMP.Salary < 30 K
And EMP.D# = DEPT.D#
And DEPT.Name is not Security
```

我们使用一些简单的示例进行查询修改。详细的算法可以在本附录参考文献［DWYE87］和［STON74］中找到。高级算法如图 C.9 所示。

查询修改算法

输入：查询、安全约束

输出：已修改的查询

对于与查询相关的约束，请通过否定修改查询的where子句

例如：如果不应该向Jane发放薪水，并且如果Jane从雇员那里请求信息，则修改查询从而从属性中没有工资的员工中检索信息

重复该过程，直到处理所有相关的约束

最终的结果是修改后的查询

图 C.9　查询修改算法

C.3.3　自主安全性和数据库功能

在 C.3.2 节中，我们讨论了查询修改，实际上是在查询操作期间处理安全约束。查询优化也将受到安全约束的影响。也就是说，一旦查询被修改，就必须构建查询树。这里的思想是在查询树中推送选择和投影，稍后执行联合操作。

其他功能也受安全约束的影响。这让我们考虑到事务管理。Bertino 等人已经开发了用于事务管理的完整性约束处理的算法（参见本附录参考文献［BERT89］）。我们已经检测了他们在事务管理中强制性安全约束处理的技术。这些技术可以适应任意安全约束。目的就是确保在事务执行期间不会违反约束。

约束可能会在元数据上执行。例如，可以授予和撤销用户对元数据关系的访问。元数据中的任意安全约束都能够以与处理数据相同的方式进行处理。

其他数据库功能包括存储管理。存储管理中的问题包括开发适当的访问方法和索引策略。我们需要审查安全约束对存储管理功能的影响。也就是说，是否可以根据约束对关系进行划分，并以这样的方式存储它们，以便能够有效地访问关系？我们需要为数据库系统开发安全的索引技术。

通过审计数据库，来确定是否发生了任何安全冲突。此外，为了安全起见，已经使用视图来授予个人访问权限。所以我们需要有效的技术进行审计以及视图管理。

在本节中，我们已经审查了安全性对一些数据库中主要功能的影响，包括查询管理、事务处理、元数据管理和存储管理。我们还需要调查安全性对其他功能的影响，如完整性约束处理和容错计算。图 C.10 所示为安全性对数据库功能的影响。

C.3.4　策略的可视化

策略实施有 3 个方面。一个是策略规范，另一个是策略实施，第三个是策略可视化。策略可视化对于复杂的安全策略特别有用。

许多应用程序都需要可视化工具，包括地理空间应用程序以及基于 Web 的应用程序，以便用户能够更好地了解数据库中的数据。可视化对于集成安全策略也很有用。例如，如果要合并来自多个组织中的多个系统，就必须将其策略可视化和合并，以便管理员能够了解集成策略。图 C. 11 所示为策略集成的可视化。

策略可视化也有助于动态策略。也就是说，当政策经常变化时，可视化的效果在设计安全系统方面将是非常有用的。在我们的一些工作中，我们使用图形结构来指定约束而不是简单的规则。这是因为图形使规则变得可视化。此外，策略可以彼此链接，并且使用图形结构可以分析各种链接以获得策略之间的关系。

数据库安全功能

查询处理：在查询处理过程中强制执行访问控制规则、推理控制，考虑查询优化的安全约束

事务管理：在事务执行期间检查安全约束是否得到满足

存储管理：开发考虑到安全约束的特殊访问方法和索引策略

元数据管理：强制实施对元数据的访问控制，确保通过发布元数据不会向未经授权的个人公布数据

完整性管理：确保在执行安全性的同时维护数据的完整性

图 C. 10　安全性在数据库功能上的影响

策略集成的可视化

语义数据模型来表示应用程序、安全约束和通过推断检测安全违规行为

应用可视化工具来检查策略的一致性

示例：Jane可以访问关系EMP里面的工资值，同时Jane还没有对EMP的读取权限。使用颜色来表示Jane能和不能访问的数据。如果数据元素具有与之相关联的两种颜色，则会有一个冲突。

使用超媒体系统浏览安全策略

图 C. 11　策略集成的可视化

策略可视化领域是一个相对比较新的研究领域。以下是 DARPA 在策略可视化方面的一些研究项目。这个领域需要做大量工作，特别是网络环境下的策略可视化，因为组

织之间需要协作并开展电子商务。策略可视化对于国土安全应用也很重要，各个机构必须共同合作、共享信息，但是也要保持自主性。

C.3.5　原型和产品

我们现在讨论在 System R 和 Oracle 中实现自主安全性。需要注意到，System R 是一个原型，而 Oracle 是一种产品。两者都是基于关系模型。有几个原型和产品都实现了自主访问控制，其中一些如图 C.12 所示。

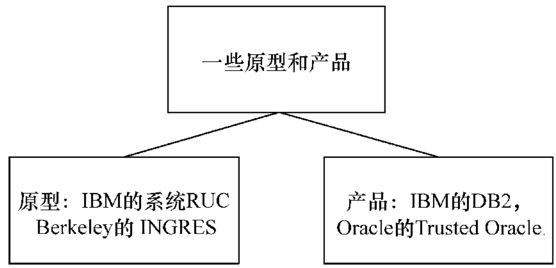


图 C.12　原型和产品

值得注意的是，随着技术的进步，原型和产品信息将不断变化。因此，在许多情况下，关于原型和产品的信息可能很快就过时了。我们讨论原型和产品的目的是解释这些概念。关于原型和产品的最新信息可以从提供商处获得，也可以从网络获得。

System R 是首先引入各种自主安全概念的系统之一（参见本附录参考文献［GRIF76］）。在此系统中，要保护的对象由表和视图表示。主体可以对安全对象强制执行若干特权。模型支持的特权包括选择（选择元组）、更新（修改元组）、插入（添加元组）、删除（删除元组）和撤销表（删除表）。该模型还支持分级管理设施，主体可以授予它所拥有的权限给其他主体。该模型还执行递归撤销。也就是说，当主体 A 撤销对表 B 的授权时，则 B 反过来将对其先前被授权访问的表的授权进行撤销。

System R 模型已经在许多方面得到了扩展。第一，组管理，其中可以授权和撤销用户组的访问权限。第二，分布式数据库管理，对于作为 System R 的分布式版本 System R^*，对授权进行了扩充。第三，否定授权。需要注意的是，System R 对安全性的大量研究已经移植到 DB2 商业产品中。有关 System R 授权模型及其扩展的详细讨论可以在本附录参考文献［FERR00］中找到。

在 Oracle 数据库服务器中，可以向用户或角色授予特权。角色是分层次组织的。一个角色获取层次结构中较低位置的所有权限。用户可以被授权担任多个角色，但是有一个限制是可以在给定时间启用或禁用角色。对于每个角色，都可能会分配一个密码，以确保仅授予对该角色的授权使用权限。

权限可以分为两类：系统权限和客体权限。系统权限允许主体对特定类型中客体执行系统范围的操作。系统权限的示例是删除数据库中任何表的元组或创建集群的权限。客体有限允许主体对特定客体执行特定的操作，这里的示例包括从特定表格中插入或删除元组。本附录参考文献［FERR00］中详细讨论了其他问题，如级联权限和特权撤销。

C.4　总结和展望

在本附录中，我们概述了数据库系统中的自主安全策略。在本附录的一开始，我们

讨论访问控制策略，包括授权策略和基于角色的访问控制。然后，我们讨论了管理策略。我们简单地讨论了识别和认证。接下来，我们讨论了审计问题以及安全视图。再次，我们讨论了策略执行。策略执行中的重大问题是策略规范、策略实施和策略可视化。我们还讨论了用于指定策略的 SQL 扩展，并对查询修改进行了概述。我们还介绍了如何将策略可视化用于集成多个策略。最后，我们讨论了一些实现自主安全性的原型和产品。我们主要关注的是关系数据库系统。

我们还有很多工作要做。例如，对于像数字图书馆和语义 Web 等新兴技术，基于角色的访问控制仍然需要做大量工作。我们需要管理策略来管理 Web 环境中的多方交易。我们还需要用于认证用户的生物识别技术。数字身份正在成为一个重要的研究领域，特别是云系统。

安全策略的实施是一个随着新技术的出现，而不断发展的话题。我们已经从关系到对象、多媒体、网络发展到云数据管理系统。每个系统都具有可以并入安全策略的一些独特功能。各个系统的实施策略将继续是一个重要的研究焦点。我们还需要对政策的一致性和完整性进行研究。策略可视化可能对此研究提供很大帮助。

云中的策略管理是一个活跃的研究领域。我们的工作包括访问控制，以及云中的基于策略的信息共享。在本书中也讨论了我们在云中安全策略实施方面开发的实验系统。

参考文献

[BERT89] Bertino, E. and D. Musto, Integrity constraint processing during transaction processing. *Acta Informatica*, 1988, 26(1-2), 25–57.

[BERT02] Bertino, E. et al., Access control for XML documents. *Data and Knowledge Engineering*, 34(3): 2002.

[CARM04] Carminati, B. et al., Security for RDF. *Proceedings of the DEXA Conference Workshop on Web Semantics*, Zaragoza, Spain, August, 2004.

[DATE90] Date, C., *An Introduction to Database Systems.* Addison-Wesley, Reading, MA, 1990.

[DWYE87] Dwyer, P. et al., Multilevel security for relational database systems. *Computers and Security*, 6(3): 1987, 252–260.

[FERR00] Ferrari, E. and B. Thuraisingham, Secure database systems. In *Advances in Database Management* (Eds. M. Piatini and O. Diaz), Artech House, UK, 2000.

[GRIF76] Griffiths, P. and B. Wade, An authorization mechanism for a relational database system. *ACM Transactions on Database Systems*, 1(3): 1976, 242–255.

[PARK04] Park, J. and R. Sandhu, The UCON usage control model. *ACM Transactions on Information and Systems Security*, 7(1): 2004, 128–174.

[SAND96] Sandhu, R. et al., Role-based access control models. *IEEE Computer*, 29(2): 1996, 38–47.

[SQL3] en.wikipedia.org/wiki/SQL, *American National Standards Institute*, Draft, Maynard, MN,1992.

[STON74] Stonebraker, M. and E. Wong, Access control in a relational data base management system by query modification. *Proceedings of the ACM Annual Conference*, ACM Press, NY, 1974.

[THUR87] Thuraisingham, B., Security checking in relational database management systems augmented with inference Engines, *Computers and Security*, 6(6): 1987, 479–492.
[THUR89] Thuraisingham, B. and P. Stachour, SQL extensions for security assertions. *Computer Standards and Interface Journal*, 11(1): 1989.
[THUR93] Thuraisingham, B., W. Ford, and M. Collins, Design and implementation of a database inference controller, *Data and Knowledge Engineering Journal*, 11(3): 1993, 5–14.
[THUR05] Thuraisingham, B., *Database Security*. Integrating Database Systems and Information Security, CRC Press, 2005.

附录 D　信息共享保障的生命周期

D.1　概述

本章介绍我们保障信息共享（ Assured Information Sharing，AIS）的方法。该研究正在由空军科学研究室（Air Force Office of Scientific Research，AFOSR）资助的 MURI（多大学研究计划，Multi – University Research Initiative）项目下进行。我们项目的主要目标是定义、设计和开发出一个信息共享保障生命周期（Assured Information – Sharing Life cycle，AISL）系统，从而实现国防部信息共享价值链。在本附录中，我们将讲述国防部面临的问题和我们开发 AISL 系统的解决方案。

国防部对 AIS 的愿景是“通过在信息环境中具有可操作性的敏捷企业，提供信息的力量，来确保任务的成功”。在目前的 AIS 项目中，我们的目标是通过定义 AIS 生命周期和开发框架来实现这一愿景。

我们项目的主要目标是定义、设计和开发出一个实现国防部信息共享价值链的 AISL。为了实现这一目标，我们正在开发以下的工具和技术：①一个综合的策略框架，通过各种策略（包括机密性、绩效制和信任）来指定和推理，以提供支持；②基于事件的安全将支持 AIS 服务的 SOA；③提供实施面向生命周期的应用和管理策略所需的服务中的安全基础设施；④利用社交网络建立信息流动的技术；⑤确保信息集成、分析和质量的方法；⑥基于行为来激励信息共享保障的工具。

本附录的组织结构如下：在 D.2 节，我们讨论存在的问题；在 D.3 节中，我们讨论 AISL 框架，第 D.4 节我们将讨论信息共享的激励措施；在 D.5 节中总结本附录内容。目前，6 所大学正在参与这个项目，他们是马里兰大学、巴尔的摩计数、普渡大学、UTD、伊利诺伊大学香槟分校、密歇根大学和得克萨斯大学圣安东尼奥分校。

D.2　问题

为了打赢全球反恐战争，国防部、联邦机构、联盟成员和首要响应者等必须主动共享信息和做出有效决定。但是，在这样做出行动的时候，人们必须保护敏感信息的机密性，并适当地尊重个人的隐私。传统的安全策略通常基于“需要获知”的概念，并且被预先定义和经常由僵化的规范所划分，其中主体和角色被授权访问哪些信息。9/11 委

员会（见本附录参考文献［MARK03］）给出的建议之一就是要从传统的角度，转向强调“需要分享”的方法。我们针对上述问题的研究将以下面信息为指导：①国防部的信息共享战略；②与国防部和其他政府机构的信息共享需求相关的情景。信息包括上下文数据、元数据，以及架构、资源、策略、流程和策略的描述。

2007 年 5 月，国防部首席信息官发表了一份阐述国防部信息共享战略的文件（DoD 2007）。信息共享的愿景是“通过具有可操作性的敏捷企业在信息环境中，发展信息的力量，来确保任务成功”。为了实现这一愿景，国防部制定了以下四个目标：“促进 、鼓励和激励共享”；“实现企业扩展”；“加强敏捷性，以适应意料之外的伙伴和事件”；“确保组织之间的信任”。国防部表示，通过在工作中执行的 5 个实施策略，将实现 4 个信息共享目标。

我们的初始方案与分布式公共地面系统（Distributed Common Ground System，DCGS）有关。为了确保联合情报、监视和侦察的横向集成（Intelligence，Surveillance，and Reconnaissance，ISR），用于改善临界时间目标的传感器平台，国防部正在以基于网络为中心，企业 SOA 将 DCGS 作为全球情报共享网络进行开发。虽然空军正在开发 DCGS（雷神公司作为主承包商），但海军正在开发其名为 DCGS－N 的版本，而陆军也正在开发其名为 DCGS－A 的版本。三个组织必须通过 DCGS，以及与外国情报部门共享战斗行动信息（NRC 2006）。我们将展示我们的研究如何能够实现和增强这一系统，并且还可以通过我们的合作者来探索其他的情景。我们的工作细节在本附录参考文献［ISI09］中给出。

D.3 信息共享保障的生命周期

AISL 由图 D.1 所示的 3 个主要阶段组成：①信息发现和广告；②信息获取、发布和整合；③信息使用和控制。这些阶段将实现（DoD 2007）信息共享的价值链。在发现阶段，各方宣传他们所拥有的信息，并搜索相关信息。信息环境中的各方可能扮演两个主要角色：信息提供者和信息获得者。信息发现和广告涉及几个问题：确定信息是什么和宣传对象是谁？支持广告有可选择性、确保机密性和检验诚信，以及确定信息共享的激励措施。信息获取、发布和整合涉及几个问题：确定要发布的信息和向谁发送信息，验证信息的需求，评估获取/发布信息的风险和后果，以及说明和评估信息获取和发布所产生的义务。在使用和控制阶段，一个关键的要求是信息提供者保持“确认信息在哪里、怎么样被使用的”（DoD 2007）。这也需要解决几个问题：如何控制已经发布

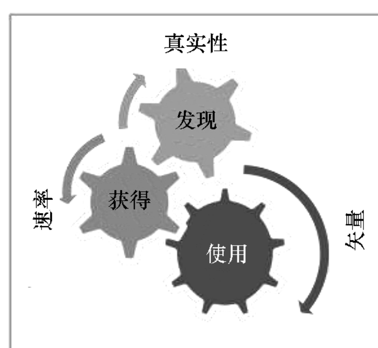


图 D.1 AISL.（信息共享保障的生命周期，IEEE 智能与安全信息会议论文集，达拉斯，得克萨斯州．©（2009）IEEE.）

的信息的使用、联合管理、访问控制和绩效，调查机密性和完整性违规，以及评估其使用其所带来的益处。

AISL 是体现信息共享保障价值链概念的具体实施方案，它提供了一套服务、工具和过程，能够共同地安全集成信息源的正确组合，并且也能够立即安全地通信、协调和适当回应现状。因此，AISL 具有高度动态性，可以对情况进行快速反应，同时提供安全保证。针对不同类型的信息，以上 3 个阶段被若干过程执行多次。图 D. 2 所示为实现 AISL 的各种模块。组成模块包括：支持 Web 服务的策略、基于语义事件的面向服务的安全架构（SSE－SOA），以及支持执行策略的安全基础设施。高级网络服务包括信息管理安全、知识管理安全、社交网络安全、激励管理安全和联合管理安全。

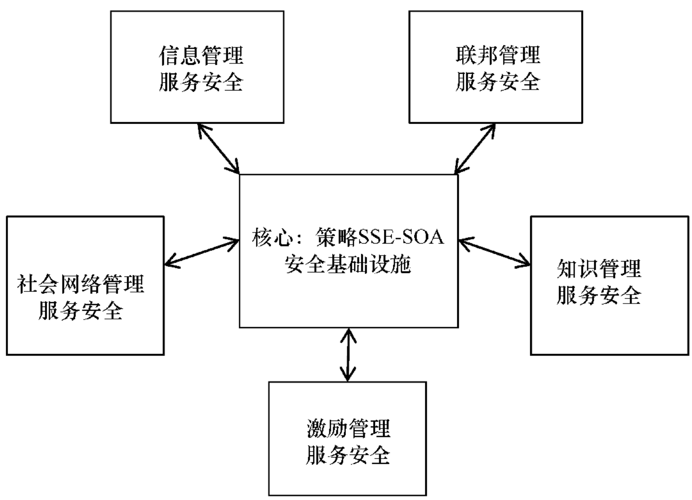


图 D. 2　服务安全（信息共享保障的生命周期，IEEE 智能与安全信息会议论文集，达拉斯，得克萨斯州．©（2009）IEEE.）

我们的目标是为决策者提供正确的信息，以便他/她可以在不确定和意料之外的情况下做出决定。我们正在开发用于评估和/或快速重新验证用于决策信息的工具和技术，并通过使用问责流程来强化关于信息的来源的管理。通过使用我们的工具，决策者可以确定信息的来源。这种意图可能会促使对信息需要进行更多的快速验证，从而快速搜索其他可以确认或初始无效的信息。AISL 是动态的，在各个阶段，都需要信息质量和来源技术来支持。

D. 4　激励和信息共享

AISL 项目为制定信息共享的策略、模式和技术做出了许多贡献。我们在这个项目中所做的独特贡献之一就是基于激励的信息共享。

我们已经建立了机制，激励个人/组织进行信息共享。一旦建立了这样的机制，我们可以使用契约理论中的概念来确定适当的奖励。例如，排名，或者在某些情况下，外

国合作伙伴给出的货币利益。首先，我们探讨如何利用分布式安全审计日志，在可靠的合作伙伴之间，对各个组织进行排名。为了处理不可能进行审计的情况，同时也为了从合作伙伴中提取信息，我们制定了游戏理论策略。我们根据经济理论进行研究，将相关结果集成到激励性的 AIS。

将现实世界中的复杂信息共享问题进行建模，转化为形式分析的过程中存在风险，是不切实际的假设。通过借鉴心理学和相关补充决策科学的见解，我们考虑了更广泛的行为假设，旨在改善系统与其预期用户之间的接口。系统的目的是整合众多信息来源，并提供各种定量输出，以帮助提高监控系统的性能。最重要的是，当信息被滥用的概率超过预设阈值时，则发送否定警报。系统整体表现的质量最终将取决于人类如何使用它。最近几十年出现的行为经济学，从心理学借鉴了如何建立具有更多现实主义经验的模型，基础假设是决策者得出推论和采取行动的方式。我们扩大了对系统激励信息共享部分的形式分析，更广泛地考虑了动因，包括人际比较，它能作为系统地塑造行为结果的因素，因此影响信息共享系统的性能。我们的结果报告在本附录参考文献［NIX12］中报道。

D. 5 总结和展望

该项目做出了几项新的贡献，并已在本附录参考文献［AISL］中报道。这些包括安全模型、框架、架构、信息管理、社交网络和对信息共享的激励。

在我们的工作中，我们将研究成果集成到云中，开发基于云的 AIS 系统。第七部分对系统做了报告。除了增强我们的系统，我们计划将与 DARPA（国防高级研究计划局，Defense Advanced Research projects Organization）或 IARPA（情报高级研究计划局，Intelligence Advanced Research Projects Activity）来共同讨论我们的项目，以开发 AIS 的现实操作系统。

参 考 文 献

[AISL] Assured Information Sharing Lifecycle: http://aisl.umbc.edu/

[CYBER] The White House National Security Council: http://www.whitehouse.gov/cybersecurity

[DoD07] DoD (Department of Defense) Information sharing strategy. May 2007, http://www.defenselink.mil/cio-nii/docs/InfoSharingStrategy.pdf

[ISI09] T. Finin A. Joshi, H. Kargupta, Y. Yesha, J. Sachs, E. Bertino, L. Ninghui, C. Clifton et al., Assured information sharing life cycle, *Proceedings of the IEEE Intelligence and Security Informatics Conference*, Dallas, TX, June 2009.

[MARK03] Creating a Trusted Information Network for Homeland Security. The Markle Foundation, 2003.

[NIX12] R. Nix, M. Kantarcioglu, Incentive compatible privacy-preserving distributed classification. *IEEE Trans. Dependable Sec. Comput.* 9(4): 451–462, 2012.

致　谢

在这里我要感谢得克萨斯大学达拉斯分校埃里克乔森工程与计算机科学学院的行政部门，让我有机会指导网络安全研究和教育中心。感谢我的同事和学生们对我提出的许多见解。特别要感谢我的同事 Latifur Khan 博士和 Murat Kantarcioglu 博士以及我在 UTD 的博士生 Vaibhav Khadilkar，因为他们关于云安全的研究和讨论，给了我编写本书的许多参考。我也要感谢项目协调员 Rhonda Walls 女士对本书的各个章节所做的校对和编辑工作。

同时我也要感谢在云计算安全方面支持我的许多人。

■ 感谢空军科学研究办公室的 Robert Herklotz 博士对我们关于云安全计算的研究提供的资金支持。没有这个支持，我永远无法获取编写本书的知识。

■ 感谢国家科学基金会的 Victor Piotrowski 博士对我们在保证云计算方面的能力建设工作上提供了资金支持。

■ 感谢得克萨斯大学的同事 Kevin Hamlen 博士、林志强博士、Kamil Sarac 博士和 I – Ling Yen 博士对云计算安全进行了讨论。

■ 感谢我们在皇家学院、伦敦大学、意大利英苏布里亚大学的各位合作者对我们在基于云的信息共享保证方面做出的努力，特别地，我想感谢皇家学院已故的 Steve Barker 博士和意大利英苏布里亚大学的 Elena Ferrari 博士和 Barbara Carminati 博士。

■ 感谢我们在云计算安全方面的合作者们，特别是已故的 James R. Johnson 博士（Bob 博士）和 ADB 咨询公司的 Anita Miller 女士以及普渡大学的 Elisa Bertino 博士。

■ 感谢加州大学伯克利分校的 C. V. Ramamoorthy 教授的鼓励与支持。

■ 感谢我之前所带的学生 Tyrone Cadenhead 博士，感谢他宝贵的评论意见。

■ 感谢我的学生们对本书内容所做的技术贡献：Vaibhav Khadilkar 先生（第 14、15、22、24、27、28、32 章）、Satyen Abrol 先生（第 14 章）、Jyothsna Rachapalli 女士（第 29 章）、Pranav Parikh 先生（第 24 章）、Arindam Khaled 先生（第 23 章）、Anduleep Ifthekar 先生（第 33 章）、Tyrone Cadenhead 博士（第 27、28 章），Mehedy Masud 博士（第 25 章）、Farhan Husain 博士（第 13、23、26 章）和 Neda Alipanah 博士（第 15 章）。

■ 感谢我的丈夫 Thevendra 一如既往地支持我的工作，我的儿子 Breman 是一个如此美好的人，感谢他们对我的激励。

Developing and Securing the Cloud/by Bhavani Thuraisingham/ISBN: 978-1-4398-6291-9.

Copyright © 2014 by Taylor & Francis Group, LLC.

Authorized translation from English language edition published by CRC Press, part of Taylor & Francis Group LLC; All rights reserved; 本书原版由 Taylor & Francis 出版集团旗下, CRC 出版公司出版, 并经其授权翻译出版。版权所有, 侵权必究。

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale throughout Mainland of China. No part of the publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher. 本书中文简体翻译版授权由机械工业出版社独家出版并限在中国大陆地区销售。未经出版者书面许可, 不得以任何方式复制或发行本书的任何部分。

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal. 本书封面贴有 Taylor & Francis 公司防伪标签, 无标签者不得销售。

北京市版权局著作权合同登记　图字: 01-2015-2209 号。

图书在版编目(CIP)数据

云计算开发与安全/(美) 巴瓦尼·杜莱辛甘 (B. Thuraisingham) 著; 林果园等译. — 北京: 机械工业出版社, 2018.6

书名原文: Developing and Securing the Cloud

ISBN 978-7-111-59831-2

Ⅰ. ①云…　Ⅱ. ①巴… ②林…　Ⅲ. ①云计算-研究　Ⅳ. ①TP393.027

中国版本图书馆 CIP 数据核字 (2018) 第 088262 号

机械工业出版社 (北京市百万庄大街 22 号　邮政编码 100037)
策划编辑: 刘星宁　责任编辑: 朱　林
责任校对: 樊钟英　封面设计: 马精明
责任印制: 张　博
唐山三艺印务有限公司印刷
2018 年 6 月第 1 版第 1 次印刷
169mm×239mm · 31.25 印张 · 695 千字
0 001—3000册
标准书号: ISBN 978-7-111-59831-2
定价: 139.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务	网络服务
服务咨询热线: 010-88361066	机 工 官 网: www.cmpbook.com
读者购书热线: 010-68326294	机 工 官 博: weibo.com/cmp1952
010-88379203	金 书 网: www.golden-book.com
封面无防伪标均为盗版	教育服务网: www.cmpedu.com

本书特色

目前市面上关于云计算开发的图书琳琅满目，但是它们大多是阐述开发理论和技术，没有涉及安全。关于云计算安全的书籍又过多地偏重安全，没有考虑到云计算本身开发利用的实际情况。本书是一本不可多得的好书，它很好地解决了这个问题，是一本将云开发与安全紧密结合、理论与实践相互融合的书籍。

作为云计算开发与安全的优秀之作，本书有以下几个特色：

- 内容全面：全书分为9个部分共35章，覆盖了云计算开发与安全的各个重要方面。
- 易于理解：书中所提及的原理和概念都有详细解释，并配有很多实例和插图帮助读者理解，以充实的内容在抽象概念和实际实现之间架设了桥梁。
- 内容新颖：不仅加入了快速发展的云计算开发和安全的新概念，还将作者团队在云计算开发与安全方面实现的几项开创性工作纳入其中。

综合来看，本书具有内容新颖、全面、实用性和指导性强等特色，不仅适合云计算开发与安全等领域的从业人员阅读，也适合作为高等院校相关专业师生的参考书。

为中华崛起传播智慧
地址：北京市百万庄大街22号
邮政编码：100037

电话服务
服务咨询热线：010-88361066
读者购书热线：010-68326294
010-88379203
网络服务
机工官网：www.cmpbook.com
机工官博：weibo.com/cmp1952
金书网：www.golden-book.com
教育服务网：www.cmpedu.com
封面无防伪标均为盗版

策划编辑◎刘星宁 / 封面设计◎马精明

尽管云计算平台和应用的使用程度迅速扩大，但大多数关于这一主题的著作都集中在高层次的概念上。长期以来一直缺少一本为如何开发安全的云提供详细指导的图书。

《云计算开发与安全》提供了云计算技术的全面概述，并填补了这方面的空白。它提出了关于如何开发和保护云计算平台和Web服务的分步指导，深入浅出地论述了相关支持技术。

本书介绍了实现安全的云计算框架以及云服务支持技术，如Web服务和安全性。同时也详细介绍了云计算框架的各个层次，包括虚拟机监控和管理程序、云数据存储、云数据管理和虚拟网络监控。它还给出了云产品和原型的几个实例，包括私有云、公有云和美国政府云。

本书回顾了云计算的最新发展，阐述了开发和保护当前云计算平台和应用的基本概念、问题和挑战；审查验证了作者及其团队在得克萨斯大学达拉斯分校开发的云计算实验系统上构建的原型。

本书适合工业、政府和学术界的读者阅读。通过本书，技术人员将会领会怎样为特定的云应用选择合适的工具；开发人员将会发现实施云的更多设计方案；管理人员将会理解最好的方案是建立自己的云还是外包出去。

ISBN 978-7-111-59831-2

上架指导 计算机／网络技术／云计算

机械工业出版社微信公众号

E视界
传播电类内容提升专业知识

科技电眼
关注电类行业动向 聚焦前沿科技

CRC Press
Taylor & Francis Group

ISBN 978-7-111-59831-2

9 787111 598312 >

定价：139.00元